

• Early Reviews: OS/2 Warp Connect
• Spotlight: Power PC Tools

September/October 1995

**Moving
to the
PowerPC!**

OS/2 Developer

THE MAGAZINE FOR ADVANCED SOFTWARE DEVELOPMENT

**Porting to the
PowerPC**

**An OS/2-Based
Airline
Reservation
System**

**Product Review:
Visual SlickEdit
for OS/2**

**Using
OpenDoc as
a "Killer"
Application
Network**

**PowerPC
Explosion!**

**BUYERS GUIDE:
Database Development Tools**

U.S. \$9.95 Vol. 7 No. 5
Canada \$10.95



A Miller Freeman Publication

WHEN I WANT ADVICE ABOUT OBJECT-ORIENTED
PROGRAMMING, I'LL ASK AN EXPERT.



SO, ASK US.

You already know reusable class libraries promise a dramatic increase in productivity and efficiency for professional application developers. You also know class libraries created with one C++ compiler can't link with code created by another C++ compiler. And that libraries written in one language can't be used with client code written in another language.

But here's something you might not know: MetaWare and IBM are changing all that.

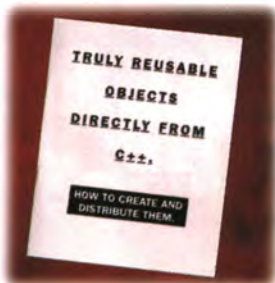
Out With The Old. In With The New. The tight binary coupling that exists between object-oriented class libraries and client code means that changes to either—no matter how small—require a complete recompilation not only of the class itself, but of all the client modules as well.

But with IBM's System Object Model (SOM) and MetaWare's High C/C++ DirectToSOM compiler for OS/2, developers can break that tight binary coupling and extend the advantages of procedure libraries to object-oriented technology.

Pretty amazing, right? And right now, you think this is where our pitch to sell you a compiler comes in. But you're wrong.

Information Only. And It's Free. We know what you really want is information. That way you can make up your own mind about the benefits of SOM and the DirectToSOM High C/C++ compiler for OS/2. We've prepared a white paper that will help. It's called Truly Reusable Objects Directly From C++: How to Create and Distribute Them. It's yours just for the asking.

To receive your free information, call, fax, or e-mail MetaWare at the numbers shown below. Once you review the facts for yourself, you'll understand why MetaWare and IBM mean "objects made easy."



CALL 408 429 6382
FAX 408 429 9273
OR
E-MAIL TECHSALES@METAWARE.COM
FOR YOUR FREE COPY.
TODAY.

 **MetaWare**
OBJECTS MADE EASY.

BEST SELLERS FROM THE Programmer's Paradise® CATALOG



Call for
a FREE
Catalog!

800-445-7899



Visual SlickEdit for OS/2— the intelligent programmer's editor by MicroEdge, Inc.



This high-powered programmer's editor is completely customizable with the speed to ignite your productivity. Save time using BRIEF, Emacs, or VI emulations. Take off with Visual SlickEdit's GUI interface, built-in dialog editor, and C-style macro language. Features include: SmartPaste™; compiler error processing; syntax color-coding; expansion and indenting; and an on-line manual. 30 day risk-free trial!

Discount Price: *219* *Offer ends October 31, 1995.
Paradise No. MEVSE3100-BE

KopyKat Remote Control by Hilgraeve



Anything you can run on a PC with OS/2 you can now do remotely via LAN, Modems, or Serial Cable. You can run: the remote computer's entire OS/2 desktop; presentation manager programs—both 16- and 32-bit; command sessions or text mode...windowed or full screen; Microsoft Windows "seamless" or full screen; and DOS sessions or programs, windowed or full screen. Runs on Novell & IBM LANs.

Discount Price: *125
Paradise No. HGKKO3100-BE

For more details
on the products
featured on these
pages call our
800 Number
and we will fax
you additional
information.



PREDITOR/2 by CompuWare Corp.

PREDITOR/2 is a graphical 32-bit source code editor for the demanding programmer. PREDITOR/2 is for programmers who don't have the time to deal with rigid programming environments, who must change capabilities to fit their exact requirements, and who need

to dramatically extend editing functions to attain peak productivity. PREDITOR/2 knows no boundaries on file size, line length, buffers, windows, undo/redo, etc.

Discount Price: *149
Paradise No. CWPDP23100-BE



Watcom VXR/REXX Client/Server 2.1 by Watcom International

A visual development environment for OS/2. Powerful connection, query and chart objects allow you to access several databases, manipulate data and chart the results quickly and easily. Features include: drag-and-drop programming; bound controls; and professional multi-threaded, multi-windowed and drag-and-drop enabled application development.

Discount Price: *281
Paradise No. WATVC3100-BE

c-tree Plus® by FairCom Corporation

DOS • WINDOWS • NT • UNIX • OS/2 • SUN • RS6000 • HP9000 • MAC • QNX • BANYAN • SCO. This well known, highly portable data management package has become established as the tool of choice for commercial development. Offering unprecedented data control, choose from direct low level access, ISAM level, or SQL access with the FairCom Server. Single User, Multi User, or optional Client/Server, ANSI Standard. Full Source. No Royalties.

Discount Price: *519
Paradise No. FACT+3100-BE



DataTable for OS/2 by ProtoView Development Corp.

DataTable is a sophisticated spreadsheet control. Through a robust message-based API and numerous notification codes, developers can control every aspect of their application. Features include: virtual memory management; column sorting; column locking; check boxes and combo boxes; and colors and fonts may be set on a cell, row; column or table basis. New for OS/2, vertical window splitting.

Discount Price: *476
Paradise No. PDDTO3100-BE



IBM® VisualAge™ C++ Version 3.0 by IBM

IBM VisualAge C++ V3.0 combines the productivity-boosting visual programming capability of IBM VisualAge with robust, professional development tools from the popular IBM C Set++™ product. This powerful

combo lets you visually build parts then assemble them into mission-critical, object-oriented programs.

CD & On-Line Docs
Paradise No. IB30H1666-BE

Discount Price: *399

Discount Price: *289

SPF/PC Version 4.0 by Command Technologies Corp.

Provides a familiar environment on the PC for ISPF/PDF main-frame programmers. Includes: Modifiable Panels & Table Services; UNDO/REDO; Program Source Colorization; SUPERC file comparison; vertical splits; mouse support; 64,000 max record length; 132-column support; and 20+ new primary commands. SPF/PC features: full 3270 compatibility; ISREDIT macros via REXX; ASCII/EBCDIC file editing; COBOL workbench integration; CUT/PASTE; fully mappable keyboard; and keyboard macros.

Discount Price: *199
Paradise No. CTSP43100-BE



MORE HOT SELLERS!

Paradise No.	Discount Price
Borland C++ 2.0 for OS/2 BOCOT8888-BE	*339
EZRAID PRO PEEPO3100-BE	*764
GammaTech Utilities STGT03100-BE	*99
IBM Smalltalk 2.0 IBM14H0270-BE	*699
IBM VisualAge 2.0 IBM17H7495-BE	*1549
Mesa/2™ ADS203100-BE	*189
MKS Source Integrity for OS/2 MKSI03100-BE	*360
OnCmd Xbase for OS/2 OLDCX3100-BE	*240
Opt Tech Sort OPOTS0100-BE	*240
Prominare Designer PRDGO3100-BE	*640
Watcom C/C++ V10.5 Competitive Upg. call for more information	*189



Phone order
800 445 7899
FAX order
908 389 9227
International Sales
908 389 9229

Programmer's Paradise®

1163 Shrewsbury Avenue
Shrewsbury, NJ 07702-4321

Call for shipping
charges/return policies.

Order Now! 800-445-7899

Circle Reader Service Number 3

In the Race for Solutions, Finish First



Introducing VisPro/Reports, the first programmable report writer for VisPro/REXX, VX REXX and OS/2 REXX

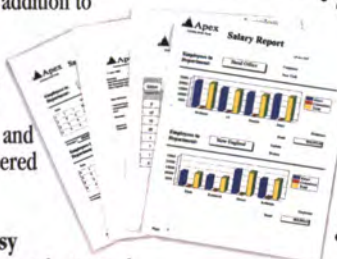
On April 28, 1993 HockWare Inc. shipped VisPro/REXX, the first OS/2 visual programming tool for REXX. Since then, we've achieved many other firsts. VisPro/REXX, VisPro/C and VisPro/C++ are the first REXX, C and C++ tools to incorporate drag and drop programming, an entity-relationship database designer, and a rich set of add-on objects including: spreadsheet, business graphics, formatted entry field, clock and calendar. Whether you're using REXX, C-Presentation Manager, or IBM's User-Interface Class Library, there is a VisPro product to meet your needs.

Yet Another First

VisPro/Reports is the latest addition to the VisPro family of OS/2 programming products. VisPro users will recognize the easy-to-use CUA'91 user-interface and the drag and drop style HockWare pioneered years ago.

Designing Reports is Easy

- WYSIWYG, free-format report design tool.



- Banded report options allow you to choose from a rich set of headers, footers and groups.
- Label reports allow you to select from a large list of Avery labels.
- Support for multiple line text, derived fields, business graphics, bitmap images, rectangles, lines and ellipses.
- Many object properties to choose from, including any OS/2 font, 16 million colors, line styles, fill patterns and drop shadows.
- Wide selection of field options, International currency and formulas.
- Numerous sample reports included.

Report Printing is Easy

- Seamlessly print or preview reports from any OS/2 REXX environment including VisPro/REXX, VX REXX and plain OS/2 REXX.
- Use your favorite REXX-enabled databases and third party libraries for access to DB2, .DBF files, EHLLAPI mainframe connections and more.
- A runtime DLL of less than 100K can be distributed royalty free.

The Complete VisPro Family

VisPro/Reports	\$199
VisPro/REXX Gold	\$299
VisPro/REXX Bronze	\$99 \$ 59
VisPro/REXX Data Entry Object Pack ..	\$149 \$ 89
VisPro/C or VisPro/C++ SPECIAL!!!	\$399 \$199
VisPro Development Suite.	\$599 \$499
(includes VisPro/REXX Gold, C and C++)	

Whether you've already discovered the VisPro family of visual programming products or you're just getting to know us, meet our latest member, VisPro/Reports. And be the first to get where you're going.

Special Offer
VisPro/C \$199
VisPro/C++ \$199
Save over 50%
Offer Expires 10/15/95

HockWare
Putting You in Control

HockWare Incorporated
P.O. Box 336
Cary, NC 27512-0336
919-380-0616
919-380-0757 FAX
Go HockWare on CompuServe
hockware@vnet.net on Internet
<http://www.nando.net/ads/hockware>

HockWare, VisPro/C, VisPro/C++, VisPro/REXX and VisPro/Reports are trademarks of HockWare Incorporated. All other company, product and brand names are trademarks and/or registered trademarks of their respective holders and are mentioned for reference purposes only. ©1995 HockWare Incorporated. All rights reserved.

IBM OS/2 Developer

SEPTEMBER/OCTOBER 1995

THE MAGAZINE FOR ADVANCED SOFTWARE DEVELOPMENT



EDITOR'S COMMENTS

PowerPC to the People 5



OS/2 TOOLBOX

Visual SlickEdit for OS/2 6



GUI CORNER

Manipulating Dialogues in C++ 10



PROGRAMMING INSIDER

Extending the OS/2 API 15



OS/2 FOR THE POWERPC

OS/2 for the PowerPC: Fact or Fiction 20

By Charles Dircks

Porting to the PowerPC 23

By Steven Tower and John Morris

Getting There From Here: Moving to the PowerPC 27

By Randell S. Flint



DATABASE DEVELOPMENT

Building an OS/2-Based Airline Reservation System 34

By Jim Monroe

Normal Form and Other Mantras of Effective Database Design 50

By Charles Meyer

A Multi-Platform Database Library 57

By Jake Jacob

Database Development Tools Buyers Guide 67



OBJECTS

OpenDoc—More Than Just Documents 38

By Robert C. Yeager



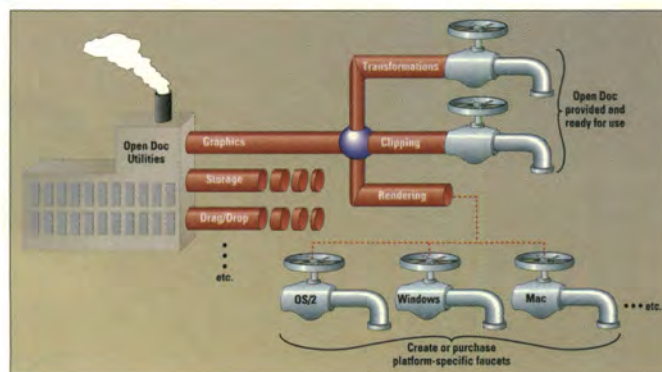
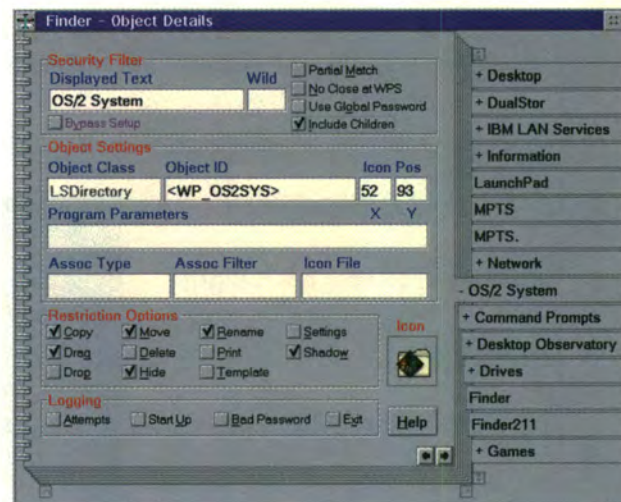
SPOTLIGHT: TOOLS FOR THE POWERPC

Tools for the PowerPC Buyers Guide 62



PRODUCT WATCH

New Products for OS/2 72

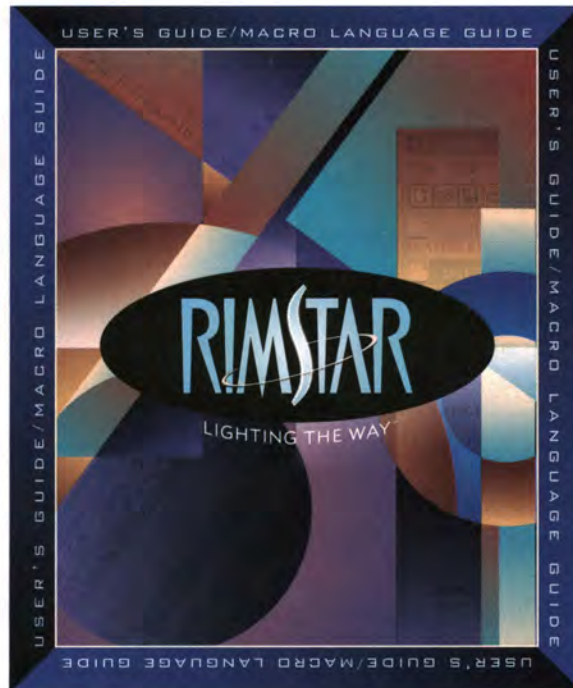


Brief and to the point:

RIMSTAR was the first native OS/2 PM editor. In fact, it was the first GUI editor of its kind. So what's *new*? and jump to errors, hex editing, and source code browser for 'C'....

Introducing RIMSTAR's

30-Day Free Trial offer. Right now, free of charge, you can use the leading OS/2 editor for 30 days – *fully functional* with our syntax coloring, complete ANSI 'C' macro language, smart C/C++ indenting and template expansion, no limits on line length, file size, number of files or windows, compile



"My copy of Brief has been permanently retired!"

Keep up the good work."A.I.

For a limited time, RIMSTAR is yours – **free** – to use for 30 days.

But why talk about features when you can try out the leading OS/2 editor for yourself – **free**? Call now for your copy: 1-800-746-7007. Outside the US, call 603-778-2500 or fax your order: 603-778-2408.

If you like it, you can have the RIMSTAR Programmer's Editor for OS/2 or Windows, Windows 95 and Windows NT for only \$199.



DESIGNED BY PROGRAMMERS FOR PROGRAMMERS

Circle Reader Service Number 4

EDITOR Dick Conklin

Compuserve: 76711,1005

MANAGING EDITOR Theresa Mori

tmori@mfi.com

EDITORIAL ASSISTANT Deborah Sommers

EDITORIAL ARTIST Peter Altenberg

VICE PRESIDENT/GROUP DIRECTOR

Regina Starr Ridley

PUBLISHER Peter Westerman

pwesterman@mfi.com

ADVERTISING SALES

Yvonne Labat

(415) 905-2353

Stephen Nikkola

(415) 905-2256

Christian O'Brien

(212) 626-2322

Kristin Morgan

(212) 626-2498

MARKETING MANAGER Susan McDonald

MARKETING GRAPHIC DESIGNER Shari Flack

ADVERTISING PRODUCTION COORDINATOR

Glenn Sadin

CIRCULATION DIRECTOR John Rockwell

CIRCULATION MANAGER Stephanie Blake

SUBSCRIPTION INQUIRIES

U.S.: (800) WANT-OS2

Foreign: (708) 647-5960

Miller Freeman
A United News & Media publication

CHAIRMAN/CEO Marshall W. Freeman

EXECUTIVE VICE PRESIDENT/COO Thomas L. Kemp

SENIOR VICE PRESIDENT/CFO Warren "Andy" Ambrose

SENIOR VICE PRESIDENT David Nussbaum

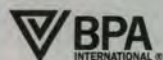
SENIOR VICE PRESIDENT H. Verne Packer

SENIOR VICE PRESIDENT Donald A. Pazour

SENIOR VICE PRESIDENT Wini D. Ragus

VICE PRESIDENT/PRODUCTION Andrew A. Mickus

VICE PRESIDENT/CIRCULATION Jerry Okabe



PRINTED IN THE U.S.A.

Editor's Comments

BY DICK CONKLIN

Power(PC) to the People

We've followed with interest IBM's quest for a portable, microkernel-based software strategy. The concept of Workplace OS has been around for two years now, but we haven't had a real life example—until now. Almost everyone agrees that this is the right way to go: an operating system personality built on a microkernel built on a hardware base. A radically new hardware base, such as the RISC-powered PowerPC, meant a new version of OS/2 was necessary. But how much work would it take to make existing applications run? Would Workplace OS preserve the OS/2 API as it had promised? Would this pave the way for some super-performing multimedia applications?

Well, the wait is over. IBM produced the operating system, first called OS/2, for PowerPC and, later, officially named OS/2 Warp Connect (PowerPC Edition). MetaWare delivered a C/C++ compiler. The main requirement was that your OS/2 application had to be a 32-bit program. Then, you simply ported over to PowerPC and . . . ?

But let's let some of our pioneering developers finish the story. We asked three of IBM's Power-

PC beta users to tell us about their migration from OS/2 Warp to the PowerPC Edition—Sundial Systems, AdeptSoft, and Pinnacle Technology. Are you ready for PowerPC? Judge for yourself.



Dick Conklin

Dick Conklin

Contact	CompuServe	Internet
Dick Conklin (Editorial)	76711,1005, or OS2DF2 Forum	os2mag @vnet.ibm.com
Miller Freeman (Circulation)	76001,2317	mjohnson @mfi.com
Peter Westerman (Publisher)	76307,1054	pwesterman @mfi.com
Theresa Mori (Managing Editor)		tmori@mfi.com

OS/2 Developer: Vol. 7, No. 4, July / August 1995

Subscription information: U.S.: One year (six issues), \$39.95. Canada, Mexico, and international surface mail, add \$16 per year for postage. Canadian GST no. 124513185. International air mail, add \$30 per year for postage. Foreign orders must be accompanied by payments in U.S. funds. For new orders and customer service, call (800) WANT-OS2 (800-926-8672) or (708) 647-5960 (fax: 708-647-5972). IBM employees and branch office customers can subscribe to OS/2 Developer through IBM Mechanicsburg's Systems Library Services (SLSS) using OS/2 Developer's order number: G362-0001. POSTMASTER: Please send address changes to OS/2 Developer, P.O. Box 1079, Skokie, IL 60076-8079. Allow eight weeks for subscriptions to begin. OS/2 Developer (ISSN 1073-0729) is published bimonthly by Miller Freeman Inc., 600 Harrison Street, San Francisco, CA 94107, (415) 905-2200. Application to mail at second-class postage rates is pending at San Francisco, CA, and additional mailing offices.

© Copyright 1995 by Miller Freeman Inc.



Visual SlickEdit for OS/2



Guy Scharf

MicroEdge's Visual SlickEdit for OS/2, version 1.72, is a powerful, graphical, programmer's editor for OS/2. The editor is extremely flexible, allowing you to easily tailor it to your own style. Language support is extensive; a version control interface is built in; color syntax highlighting is included; and a Slick-C programming language allows you to modify and extend the editor as needed. Smart syntax expansion writes some of your code for you. There's even a spell checker!

Visual SlickEdit offers a broad range of cross-platform support than most other editors. MicroEdge has versions of Visual SlickEdit for Windows 3.1, NT, and 95 as well. A related product, the character-mode SlickEdit editor, is available for DOS and more than 15 flavors of Unix.

Visual SlickEdit is supplied on three diskettes, and installation is straightforward. A complete installation takes 6.8 megabytes on your hard disk; a minimum installation requires 4 megabytes. A 25-MHz 486 is recommended as the minimum system. I tested Visual SlickEdit for OS/2 on a 50-MHz 486 with 32 megabytes of memory and found performance quite acceptable.

CONFIGURING VISUAL SLICKEDIT

When you install Visual SlickEdit, you select the keyboard emulation that you prefer to use. CUA, Brief, Emacs, SlickEdit, and vi are supported. You can change your selection later using the Configuration dialog (see Figure 1). The keyboard emulation is implemented as bindings between keys and internal functions. You can easily alter these bindings to produce any keyboard you want. All of the keystroke combinations that I refer to here are for the CUA keyboard emulation style; other emulations may use different keystrokes. The manual and online help de-

scribe the keystrokes for each keyboard emulation. You can also ask Visual SlickEdit to create a file with the active definitions.

Figure 2 shows the main editing screen. You can configure the menu, button bar, scroll bars, and more using the Configuration dialog. An unusual feature is that you can alter Visual SlickEdit's own dialogs. The Slick-C programming language includes dialog support. Since Visual SlickEdit's own dialogs are created using the same tool, you can modify them to obtain the appearance you want.

Using the File Extension configuration option, you can tailor the Visual SlickEdit's behavior based on the file extension. Figure 3 shows how I have defined the behavior for C files: for example, tabs are at intervals of four columns. Using other dialogs, you can select which programming style variation you want for syntax expansion, such as where the braces should be located after an if statement.

The dialogs shown only touch the surface of the ability to configure Visual SlickEdit. In my testing, I found minor features that did not operate the way I wanted them to. In all cases, I was able to change the configuration so that I got the behavior that I wanted. Finding the right way to request the change was not always obvious, but MicroEdge technical support was able to direct me to the solution quickly.

EASE OF USE

Visual SlickEdit is easy to use, with lots of tools to help the programmer. For safety, backups are made when a file is saved and optionally on a timed basis as well.

The editing functions are powerful. Line, block, and character selection are all available. The new SmartPaste function is especially nice: paste code into an existing function and it is automatically indented cor-



rectly for the location you are pasting it into. When you need word processing, automatic word-wrap, margins, and justification are available. Undo and redo is supported for up to 32,000 events. While that's not unlimited, that's more than I ever edit in one session.

Color syntax highlighting clearly separates comments from code and immediately shows any errors in closing a comment or string. Syntax expansion and indenting work well. Type `if` followed by a space and Visual SlickEdit inserts:

```
if ()  
{  
}
```

in your program.

Support is provided for C, C++, Pascal, Fortran, Basic, dBASE, Modula-2, Assembly, COBOL, and Ada. The manual describes how you can add support for other languages as well.

Visual SlickEdit allows you to group files together in *projects*. In a project configuration dialog, you define what files are part of the project. For each project, you can define the *make* command for the project, or the commands to compile individual modules, in a project configuration dialog shown in Figure 4. With the project defined, `Ctrl+M` compiles the project. Since this is OS/2, you can continue editing while the compiler is running. Compiler output is placed in a process buffer, and a window is opened for the buffer. Double-click on any message, and a window is opened showing the module and line in error. Press the key for *next error* or *previous error*, and you are moved to the next or previous line that generated a message.

Visual SlickEdit supports several version control systems, including TLIB and PVCS for OS/2. Using the Project menu, you can check files in or

out. I especially liked that a window is displayed for you to enter the description of the change when you check a file into PVCS. The editor doesn't simply start a command window and let you deal with the prompts—it effectively wraps a graphical front-end around the character mode version control system. As with all of Visual SlickEdit, you can configure the version control check in and out process to operate as you wish.

Tags allow you to locate and edit functions quickly. You can generate tags for individual files or for an entire project. Once the tag file has been created, position your cursor on a function name, press `Ctrl+.` and a window will be opened showing that function.

Visual SlickEdit gives quick access to OS/2 programming references. Place the cursor on a function or symbol, press `Ctrl+F1`, and the help entry for that function will be displayed. This lets you look up the parameters for `WinCreateWindow` or `fopen` quickly.

Searching capabilities are powerful. Regular expressions are supported. The search can encompass one file or many files. When you are searching multiple files, a search result window, showing all matches, is displayed. Pick the match of interest and press the *GoTo* button to open that file.

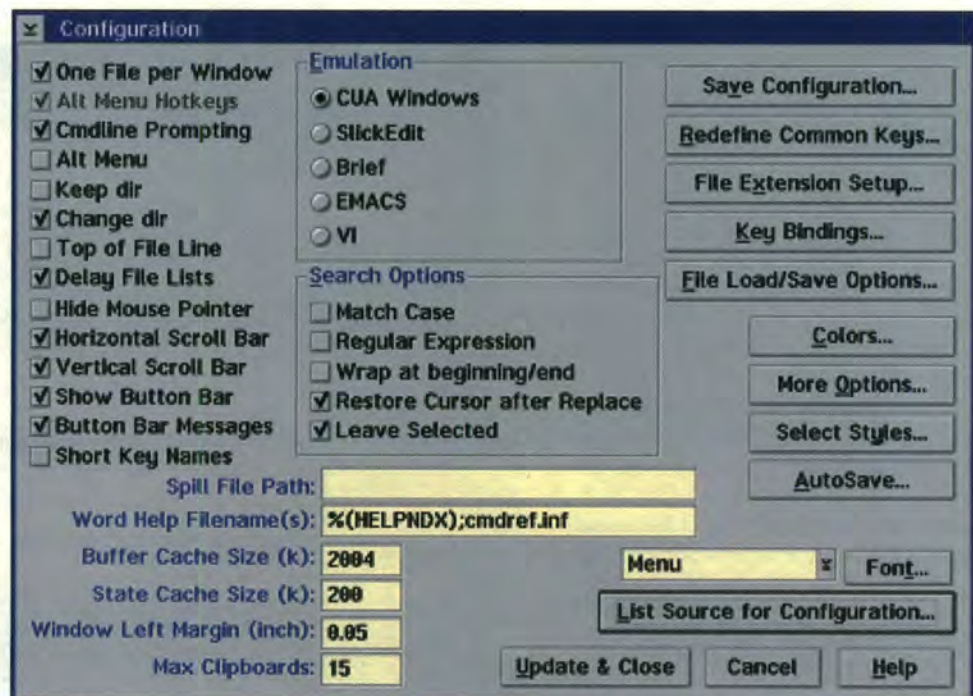


Figure 1. Configuration dialog.

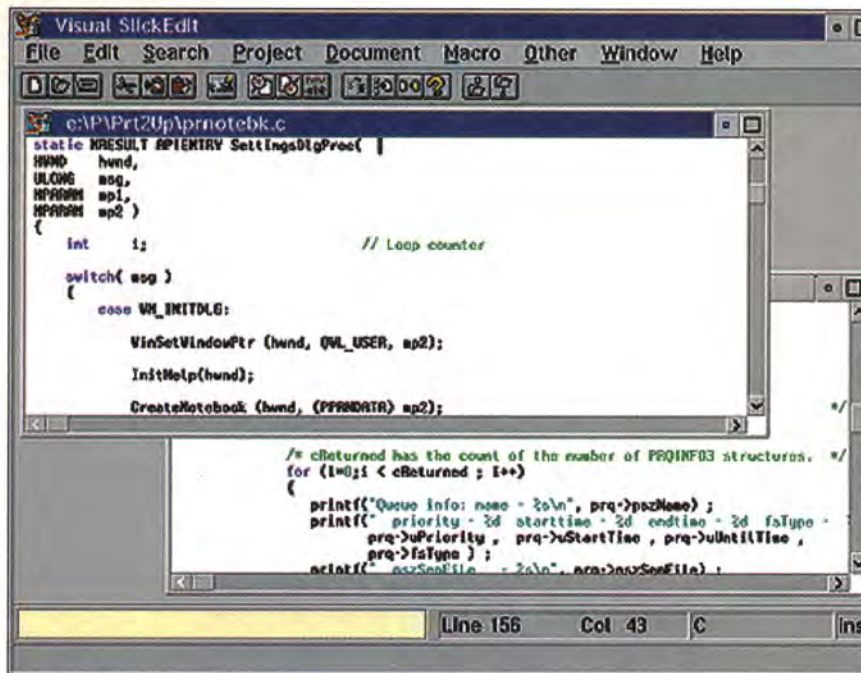


Figure 2. Main editing screen.

Incremental searching is especially convenient. Just start typing and the cursor moves to the next location in the file that matches what you have typed so far.

A calculator is included, too. The calculator supports hexadecimal, octal, and binary computations as well as decimal. Calculations can mix different representations. If you don't need a full calculator, simple expression evaluation is included on the command line. Type "math 2*2," press return, and Visual SlickEdit adds "=4" to your statement. Or type "math 0xA003," press return, and the decimal value will be displayed. The hexadecimal and decimal values are always displayed on the status line below the calculation, so you can always see both representations. This is very useful when using APIs that have error returns documented sometimes as decimal and sometimes as hexadecimal.

One of the less visible but powerful features is the *alias* facility. In its simplest form, aliases can be used for command line expansion. For example, you can define "lw" to be an alias for a fully qualified path to a file name. Select File/Open, type "lw," press Ctrl+Spacebar, and the fully qualified name will be substituted for "lw." A more complex use would be to automate code or documentation templates in a file. You could type "f," press Ctrl-Space, and create a function header for the language you are using. I plan to use this feature for documenting functions and creating skeleton programs.

Visual SlickEdit also has text and

binary file comparison, a file manager, split windows, begin/end structure and brace matching, and more.

Macro programming is the key to extending the power of Visual SlickEdit. The Slick-C macro language is a typeless, C-like language. To ease macro creation, macros can be recorded from keystrokes and saved for later editing and expansion. Macros are source and binary code compatible on all OS/2 and Windows platforms, making it easy to develop tools for cross-platform development. More than 130 macro files are included with Visual SlickEdit. These all serve as ex-

amples on how to program in Slick-C.

The macro programming language itself is straightforward. Programming it does require learning a large set of Visual SlickEdit functions. For example, this small macro inserts // at the beginning of every line in a marked block:

```
defmain()
/* Puts // in front of selected lines
*/
{
    if (begin_select()) return("");
    bl=p_line;
    end_select()
    el=p_line;
    for (i=bl;i<=el;++i) {
        p_line=i;
        begin_line();
        keyin "// "
    }
    deselect();
}
```

DOCUMENTATION

Visual SlickEdit comes with a 650-page manual of which 130 pages cover configuration and use. The majority of the manual describes the Slick-C programming language. The configuration section contains the information you need to use Visual SlickEdit effectively. However, this section is not well organized, so you may need to hunt around a bit for the information you need. I did find the table of contents useful for locating information. The documentation is clearly written and easy to understand.

The programming language is

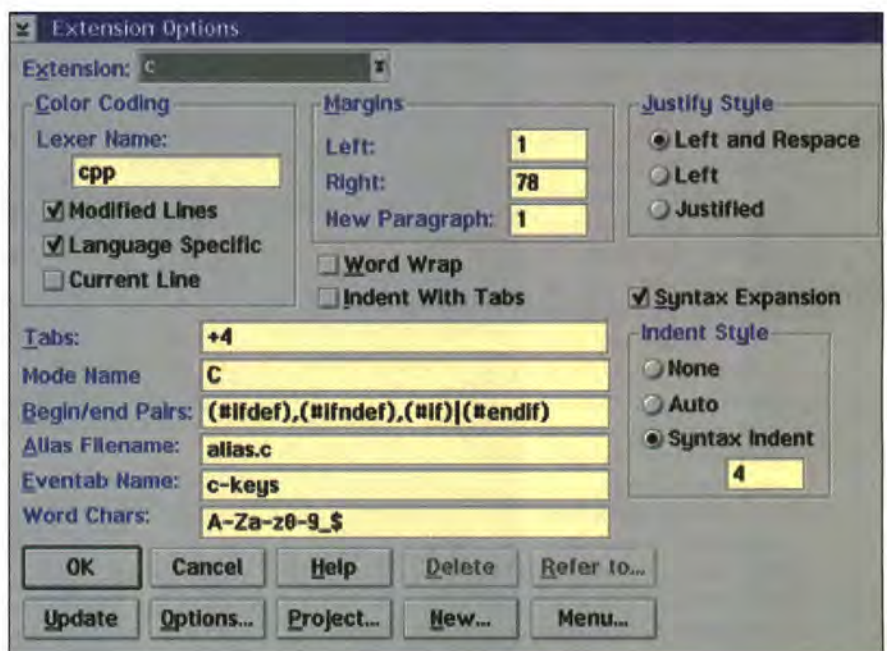


Figure 3. File extension options.

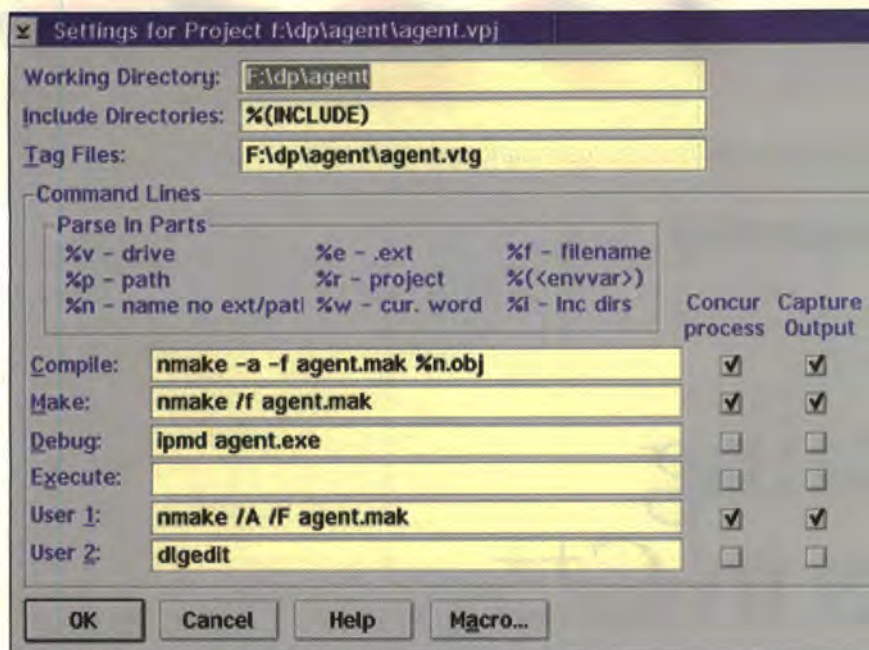


Figure 4. Project options.

reduce the risk of two people changing the same file at the same time. The second copy of the file is opened in read-only mode and is refreshed with changes made on the other copy. I saw a few odd behaviors, such as the refresh not always occurring, but overall I found the network support to be satisfactory.

TECHNICAL SUPPORT

MicroEdge supports Visual SlickEdit on CompuServe. GO SLICKEDIT and select message and library section 13, MicroEdge Inc. I found several messages posted on the forum each day. I posted several questions and reported two problems. My questions were answered the next day. Updated versions of Visual SlickEdit or macros to fix problems were uploaded to the library within a day or two. I could hardly ask for better support!

As this review was completed, MicroEdge released an update to support the IBM WorkFrame product. The update is available for downloading from their forum.

SUMMARY

I have resisted converting to a graphical editor because of my large investment in Brief macros for documenting and organizing code. With Visual SlickEdit, I may finally take the plunge and convert my macros. Visual SlickEdit is powerful, flexible, and a delight to use. It's a definite keeper for my toolbox.

Guy Scharf is president of Software Architects, Inc., a software development firm specializing in developing OS/2 Presentation Manager applications for vendors and business. He can be reached at 76702.557@compuserve.com or at Software Architects, Inc., 2163 Jardin Drive, Mountain View, CA.

PRODUCT INFORMATION

Visual SlickEdit for OS/2, version 1.72.....\$295

Vendor: MicroEdge
P.O. Box 18038
Raleigh, N.C. 27619-8038
Phone: (919) 831-0662
Fax: (919) 831-0101

described in 80 pages; the remaining 570 pages are an alphabetical reference of all of the macro functions. The language description provides a good introduction to the language and the philosophy behind it. A nice touch is a section describing common programming mistakes. These mistakes are used to illustrate how the language works. In the reference section, the macro functions are briefly described in one or two sentences. More detailed explanations and examples are included for more complex macros. Slick-C is yet another language to learn—I would have preferred REXX—but it appears to be relatively approachable. Visual SlickEdit's macro recording provides a good way to get started learning the language.

WARTS AND BLEMISHES

Visual SlickEdit for OS/2 is very similar to Visual SlickEdit for Windows and reflects a Windows user interface design. Configuration involves an extensive set of nested dialogs. As usual with nested dialogs, it is easy to forget exactly where you are. The OS/2 notebook control could organize configuration information much more effectively. Other dialogs, such as the File Open dialog, reflect a Windows design.

Visual SlickEdit uses the Multiple Document Interface (MDI) paradigm for organizing windows. All windows that you open are children of the client area of the main Visual SlickEdit window. You can arrange the windows as you wish; Window menu items will

arrange all windows as cascaded or tiled. The MDI main window reduces the maximum vertical size of each window you open, as the main window has tool bars, status lines, and more. I would prefer to see windows be children of the desktop instead, as is the conventional practice for OS/2. The user would then be able to use screen space more effectively.

While Visual SlickEdit emulates keystrokes for Brief, it will not execute Brief macros nor is a conversion utility available. You must thus convert Brief macros manually, learning the Slick-C language along the way. However, the product includes a very rich set of macros. I found that Visual SlickEdit had equivalent functions for some of the Brief macros that I rely on when writing programs.

The tag function is overly generous about identifying functions. In my testing, it "found" function definitions in comments. The only way to be certain of what is a function definition is to actually parse the file using the syntax rules of the language. This is apparently not done.

I normally set my default system font to 8.Helv rather than using the default System Proportional font. I could not do that with Visual SlickEdit. When I changed the system font, text and spacing in some dialogs were altered so much that I could not read the dialog at all. The problem was not the usual text clipping but a more extensive alteration.

Visual SlickEdit optionally locks files in a network environment to



In this issue, your feline-loving furballs investigate dialogues from a C++ perspective.

By MARK BENGE and MATT SMITH

Manipulating Dialogues in C++



Mark Benge



Matt Smith

Decisions, decisions, decisions. While investigating the possibility of providing C++ wrappers for the print API discussed in our previous three columns, we reached the conclusion that such an effort was premature in light of our abundant use of dialogues. As Mark's five-year-old nephew, Andrew, might put it, "Ding, I've got it!" A bell went off in our collective heads. What was that bell? Wouldn't it be nice to investigate and develop a set of C++ classes that have the capability of processing a dialogue template, thereby generating the appropriate C++ control classes that wrap the controls in the template? Yup! We thought so too. Therefore, our goal is to provide you with the necessary classes and implement them using the IBM Open Class Library.

THE INVESTIGATION

There are a number of ways we can tackle this assignment, with each way achieving the common goal:

- 1) Load the dialogue template using a class derived from `IFrameWindow` and use the nested class `IWindow::ChildCursor` to iterate the child controls in the dialogue. Use the `WinQueryClassName` API to determine the type of control based upon its Presentation Manager window class, subsequently creating C++ objects for each control.

- 2) Derive a new class from `IFrameWindow` and, in the constructor for this new class, load the dialogue. Provide a member function that loads the dialogue template from the resource via the `DosGetResource` API. Use the information in the dialogue template to iterate the various child controls in the template and create a corresponding C++ object for each control.

- 3) Derive a new class from `IFrameWindow` and, in the constructor for this new class,

load the dialogue template from the resource via the `DosGetResource` API. Use the information in the dialogue template to iterate the various child controls in the template and create each control using the `WinCreateWindow` API. A corresponding C++ object can be created for each control by using the window handle returned from `WinCreateWindow`.

Each solution has its merits, as well as its faults. The first solution is more of a C++ solution than the other two, as most, but not all, of its implementation is in C++. The second solution seeks to take advantage of Presentation Manager's support for dialogues, and the third solution places all the dialogue processing under your control. We chose the second solution for a number of reasons:

- Taps Presentation Manager's dialogue-processing capabilities
- More intuitive for C programmers who have dialogue backgrounds
- Potential bridge for C programmers who are seeking to transpose their work to C++
- Performance.

ON TO THE CHASE

The base class that we developed for the processing of dialogue templates, `Dialog`, is derived from `IFrameWindow`, taking advantage of some of the support that `IFrameWindow` already has for dialogues. The `Dialog` class constructor, shown in Figure 1, is implemented using various C++ classes. The dialogue is loaded by the member function, `IResourceLibrary::loadDialog`. This function uses the `WinLoadDlg` API to perform the load. If the load fails, `IResourceLibrary::loadDialog` will throw an `IAccessError` exception.

Why do we load the dialogue and dialogue template in separate steps? Can't we

load the template and create the dialogue in one step to help our performance? Nope! We can only iterate the child controls in the dialogue template to create C++ objects after the base `Dialog` class is constructed. We are calling virtual functions that may be overridden in a derived class, which means the derived class must exist, or we end up calling the virtual functions in the base class. We will discuss this approach in more detail later.

A couple of other points from the constructor in Figure 1 that are worth mentioning are the `IThread::current().initializeGUI()`, and the `IFrameWindow::start` function call. `IThread::current().initializeGUI()` initializes the Presentation Manager environment by calling the `WinInitialize` and `WinCreateMsgQueue` APIs. The `IFrameWindow::start` function creates a new frame handler and adds it to the frame window, initiates message handling for the window, and adds the window to the internal window list that is maintained by the IBM Open Class Library.

TO ITERATE OR NOT TO ITERATE

Once we have loaded the dialogue and the dialogue template, we can iterate the template and construct the corresponding C++ objects for the child controls. We use the information contained within the template to determine the number of child controls, the type of control, the individual control IDs, and the control style. Why is the control style important you may ask? You must use the style to determine the type of button and static control you are processing, because multiple button and static controls rely upon the `WC_BUTTON` and `WC_STATIC` window classes, respectively.

In our default implementation, we have chosen to iterate the template and construct C++ objects for each of the child controls, immediately following the loading of the dialogue. The dialogue is loaded in the `Dialog` constructor, and the C++ objects are constructed in `Dialog::createControlsFromTemplate`. Using the default implementation, C++ objects are constructed for each child control that has an ID other than -1; child controls with an ID of -1 are skipped. However, you can defer the construction of the C++ objects until they are needed, by simply leaving out the call to `Dialog::createControlsFromTemplate`. When you need to construct a C++ object, member functions are provided that allow you to easily create an object for a specific

child control. All you need to supply these member functions is the child control's ID. Each of these functions is identified by the "WithId" suffix. An example is `Dialog::checkBoxWithId`, as shown in Figure 2. We will discuss the elements of this function in more detail later.

During the iteration of the child controls in the dialogue template, we use the `offClassName` field within the `DLGITEM` structure to determine the type of control. (This technique is the same used in the `PDSGetTemplate` function that is found within the support.c module from previous article sources.) If the ID is not -1, we construct a C++ object using the constructor that is designed for the creation of an object from a dialogue template. Each Presentation Manager control wrapper within the IBM Open Class Library has this type of constructor that accepts a window ID and a pointer to the parent window object. After the object is created, we call the member function, `IWindow::setAutoDeleteObject`, which results in the destruction of the object when the Presentation Manager window is

```
Dialog :: Dialog( unsigned long ulWindowId,
                  IWindow* pOwner ) :
    IFrameWindow( deferCreation ),
    ulId( ulWindowId )
{
    IThread::current().initializeGUI();

    IResourceId
        resId( ulWindowId );

    IResourceLibrary
        resLib( resId.resourceLibrary() );

    IWindowHandle
        dialog = resLib.loadDialog( resId.id(),
                                    0,
                                    pOwner,
                                    WinDefDlgProc,
                                    0 );

    /* Dialog was created. Call
    IFrameWindow::start */
    /* to create a new frame handler, add it to
    this */
    /* frame window, and initiate message
    handling */
    /* for the window. */
    start( dialog );
} //Dialog::Dialog
```

Figure 1. Constructor for dialog.


```

ICheckBox* Dialog::checkBoxWithId( unsigned long ulId )
{
    ICheckBox*
        pCheckBox = (ICheckBox *)IWindow::windowWithOwner( ulId, this );

    if ( !pCheckBox )
    {
        pCheckBox = new ICheckBox( ulId, this );
        pCheckBox->setAutoDeleteObject( true );
    }

    return( pCheckBox );
}

```

Figure 2. Dialog::checkBoxWithId member function.

destroyed. Our final act is to call a protected virtual function, passing a pointer to the newly created object. The default implementation of these functions does nothing. You must override them in a class derived from Dialog to add the desired behavior. A small snippet of code shown in Figure 3 illustrates these steps in relation to the check box control. We must use the style as well as the button window class to identify the check box.

SUPPORTED CONTROLS

All of the controls that are supported by the Dialog class are listed in Table 1. The controls that are not supported are the valueset and most of the multimedia controls. The valueset is not supported since it is not yet supported by the Open Class Library. However, you can implement support for the valueset via the valueset C++ wrapper that is discussed in the book, *OS/2 C++ Class Library Power GUI Programming with C Set++*.

```

switch ( pdlg->adlgti[i].offClassName )
{
    case WINCLASS_BUTTON :
    {
        unsigned long ulStyle = pdlg->adlgti[i].flStyle &
                                BS_PRIMARYSTYLES;

        switch ( ulStyle )
        {
            case BS_CHECKBOX:
            case BS_AUTOCHECKBOX:
            {
                ICheckBox*
                    pCheckBox = new ICheckBox( ulId, this );
                pCheckBox->setAutoDeleteObject( true );
                checkBox( pCheckBox );
            }
            break;
        }
    }
}

```

Figure 3. C++ object construction.

WHAT ABOUT CUSTOM CONTROLS?

Custom controls are also supported to an extent. We detect the presence of a custom control in the Dialog class constructor by interrogating the value of the cchClassName field of the DLGITEM structure. If it is nonzero, we assume that it is a custom control and call the member function, Dialog::customControl, passing the window ID and a constant reference to the class name string as parameters. The default implementation of this function does nothing. You must override this function in a class derived from Dialog, to properly process the custom control.

DEFERRED CONSTRUCTION OR IS IT A QUERY?

In Figure 2, we show a member function, Dialog::checkBoxWithId. This function and the other "xxxXxxWithId" functions play multiple roles. You can use them for deferred construction of C++ objects or for querying an object that already exists. In Dialog::checkBoxWithId, IWindow::windowWithOwner is called to obtain a pointer to the check box object that corresponds to the passed ID. If a NULL pointer is returned, we attempt to construct a check box object, and call IWindow::setAutoDeleteObject, to allow the IBM Open Class Library to manage the destruction of the object when the check box window is destroyed. Why do we say attempt, and why do we not throw an exception on a failed attempt? If the object construction fails, the IBM Open Class Library throws an exception for us. If a non-NULL pointer is returned by IWindow::windowWithOwner, the call to Dialog::checkBoxWithId is, in essence, a query, and the pointer is returned to the caller.

SAMPLE DIALOGUE

The sample driver application il-

lustrates some of the things previously discussed. A new class, SampleDialog, is derived from our base Dialog class. SampleDialog is designed to support a sample dialog that we have cobbled together to show how to provide control interaction from a C++ perspective. As mentioned, we create C++ objects for the child controls by calling Dialog::createControlsFromTemplate. This function is called from within the SampleDialog constructor.

GETTING A HANDLE ON THE HANDLERS

Once we have the C++ objects constructed, our next step is to attach general purpose handlers to process the various window events generated by the controls. The first handler class we would like to attach is a command handler, ICommandHandler. We use an ICommandHandler to process the command, WM_COMMAND, event that is generated by clicking on the "Query" or the "Cancel" push button shown in Figure 4. In our driver application, we derive a new class, SampleCommandHandler, from ICommandHandler and override the member function, ICommandHandler::command. SampleCommandHandler::command processes the command event, and as you may notice when you preview the driver application, it bears a strong resemblance to a window procedure. SampleCommandHandler is attached to the sample dialog by calling IHandler::handleEventsFor from within the SampleDialog constructor.

The cancel processing within SampleCommandHandler::command is, as you have guessed, a dismissal of the dialogue. The query processing, although more involved, is still quite simple. We query the selected item's text within the combo box and the list box, as well as the current value in the numeric spin button. We display the text and value with the multi-line edit (mle) control.

Note that the return code for a handler has significance. A return code of false indicates that the event may be dispatched to other handlers for processing or that it may be processed by the default window procedure. A return code of true indicates that processing of the event has finished. However, if you return true, you may also be required to set an event result via IEvent::setResult. Most events do not require the setting of results, but be aware of those that do.

The next handler class we would like to attach is a select handler, ISelectHandler. We use an ISelectHandler to process the selection, LN_SELECT, noti-

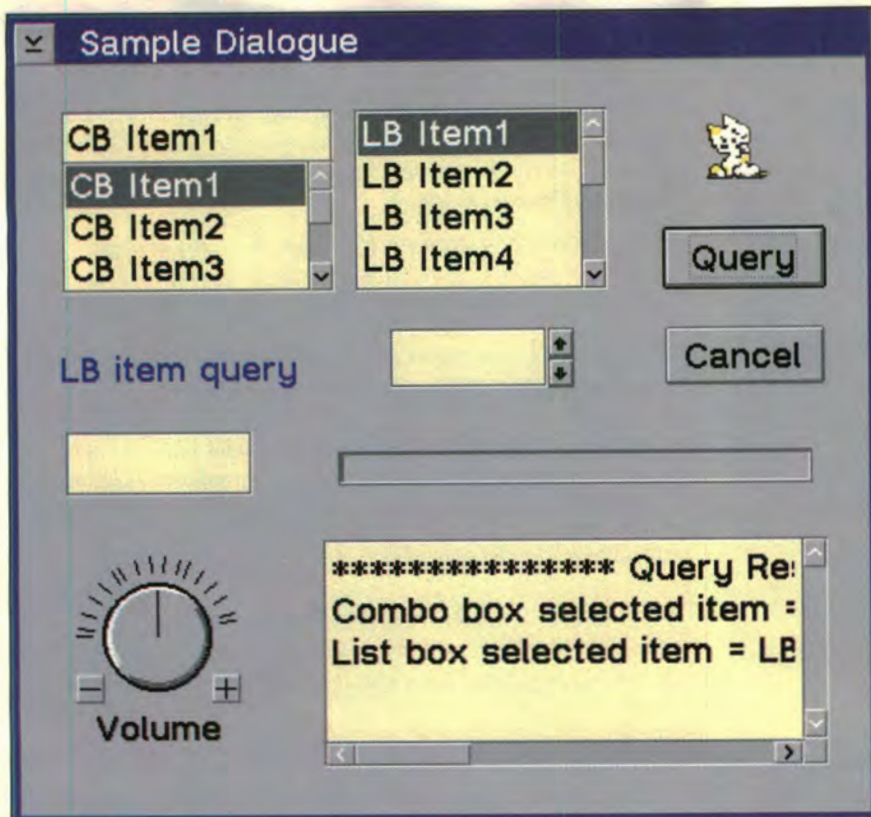


Figure 4. A handler is selected by generating an item in the list box.

fication event that is generated by selecting an item in the list box shown in Figure 4. In our driver application, we derive a new class, `SampleSelectHandler`, from `ISelectHandler`, and override the member function, `ISelectHandler::selected`. `SampleSelectHandler` is attached to the sample dialog by calling `IHandler::handleEventsFor` from within the `SampleDialog` constructor.

`SampleSelectHandler::selected` processes a selection event and writes the selected item's text into the static text control.

YOU ARE BEING OBSERVED

Most of us have dealt with the nuances of control event processing at one time or another. Since many of the

notification values overlap between controls, you must determine the control type prior to tackling the notification value. We use the notification framework in the IBM Open Class Library to remedy that headache. Rather than using integers to represent the notification event, unique strings are used instead.

We derive an observer class, `SampleObserver`, from `IObserver`, overriding the pure virtual function, `IObserver::dispatchNotificationEvent`. `SampleObserver::dispatchNotificationEvent` contains our logic for processing the `ICircularSlider::valueId` notification. We use the information that is passed by the notification event class, `INotificationEvent`, to update an entryfield,

as well as a progress indicator (a read-only slider).

We enable observer notification within `SampleDialog::circularSlider` by calling `ICircularSlider::enableNotification`. Our observer, `SampleObserver`, is attached to the circular slider in this function via `IObserver::handleNotificationsFor`. We encourage you to explore the code, as the sample driver should make things clearer. On a final note, you need to attach the observer to the control you are observing. It does not work if you attach the observer to the owner of the control.

HANDLING DIALOGUE INITIALIZATION

We've purposely avoided the issue of handling dialogue initialization. Our general purpose handlers cannot be used to process the dialogue initialization, `WM_INITDLG`, event, since we are specifying `WinDefDlgProc` as the dialogue procedure when we call `IResourceLibrary::loadDialog`. Special logic is required to enable initialization processing, and because this logic has already been implemented elsewhere, we do not want to "re-invent the wheel". An excellent implementation is provided in the book, *OS/2 C++ Class Library Power GUI Programming with C Set++* on pages 464-471.

CONCLUSION

The sample driver application, although simple in scope, illustrates some powerful concepts. A common problem faced by users of the IBM Open Class Library is how to support legacy resources from the context of a C++ class library. Hopefully, we have given you a set of classes that you can use to implement support for your legacy resources from a C++ class library.

One possible extension of the `Dialog` class is to convert a dialogue template into a canvas. As is the case for dialogue initialization, we will avoid this issue and leave it as an exercise for the brave at heart.

Please keep those calls, letters, and suggestions coming, and we will try to incorporate your suggestions into future columns.

Acknowledgements

Special thanks to Bob Love, Bruce Olson, and Hiroshi Tsuji of the IBM Open Class Library group for their design tips.

Mark Benge, IBM Software Solutions, Cary, N.C., is a staff programmer who joined IBM in

GUI Corner on the Internet

Announcing the opening of *Tails from the Trenches* home page in the *GUI Corner* web page, <http://www.prominare.com/prominare/gui-corner.tailsfromtrenches.html>. This home page is dedicated to the timely release of "kitty treats," as they become known to us. In case you have forgotten, our other home pages are <http://www.prominare.com/prominare/guicorner.html>, which is a repository for current and past code examples, and <http://www.prominare.com/prominare/guicorner.catsmeow.html>, which contains errata.

Buttons

- Check box (including auto style)
- Push button (including graphical style)
- Radio button (including auto style)
- User-defined button
- 3-State check box (including auto style)

Circular slider

Combo box

Container

Entry field

List box

Multi-line edit

Notebook

Scroll bar

Slider (including progress indicator)

Spin button (both text and numeric only)

Static

Bitmap

Group box

Icon

Text

Table 1. Supported conditions.

1989 and has worked on various CUA '91 controls for OS/2 2.x. He works in IBM user interface class library development, where he was involved in the implementation of C++ classes for drag-and-drop in C Set++ 2.1. Mark has a B.S. in computer science from Western Carolina University. He can be reached on

CompuServe at 73532,2063 and via Internet at banzai@vnet.ibm.com.

Matt Smith, Prominare Inc., Toronto, Ont., is lead architect for the Prominare Development System, an OS/2 2.x advanced GUI develop-

ment environment. Matt has been actively involved with OS/2 since 1988 and cofounded Prominare in 1990. He has a degree in architecture from the University of Waterloo and can be reached via Internet at msmith@interlog.com.

REFERENCES

Credits: Screen captures were done through OpenShutter from One UP Corp. and touched up through PaintBrush in Win-OS/2. The resource file for the driver application, driver.rc, was generated using Prominare Designer Release 4.

You can download dlccpp.exe from these electronic sources:

CompuServe's OS2DF2 forum (OS/2 Developer Mag Section).

File area 11 on the IBM PCC BBS, (919) 517-0001, OS/2 Tools section on the OS/2 BBS for IBM TALKLink customers.

FTP to the site address prominare.com and log in as anonymous. The source is located in the /pub/ prominare/guicorner-code cache subdirectory.

C/C++ Tools, User Interface Class Library Reference (S819-1179).

C/C++ Tools, User Interface Class Library User's Guide (S829-3743).

OS/2 C++ Class Library Power GUI Programming with C Set++ (SR28-5626).

Point and click your way to fast development of GUI client-server applications.

Do it with Guidelines, a second-generation object-oriented graphical workbench. Guidelines lets you create graphical client/server applications for OS/2 and Windows clients. A built-in prompter lets you simply point and click or drag and drop from the desktop using more than 30 presentation manager controls.

Guidelines utilizes JBA's high-level object-oriented language (JOT) to describe applications. After the application is designed, JOT automatically converts to C++ code, so you don't have to learn its complex syntax and rules. But if you already know C++ and want to use it directly, you can. The conversion takes place regardless of the platform intended.

Guidelines saves time and simplifies the creation of graphical, platform-independent applications. Its scalable architecture lets you create anything from stand-

alone PC applications to complex enterprise-wide business solutions. Guidelines supports DB2/2, DB2/400, DB2/6000, DDCS, ODBC, Q+E Lib, and SQL.

Call for a free demo pack: 1-800-JBA-INTL.
In Canada: (905) 940-2442

JBA International
161 Gaither Drive, Ste. 200
Mt. Laurel, NJ 08054





In this issue's "Programming Insider," we'll explore how IBM's new Developer API Extensions help you, IBM, and the programming community whether you already have OS/2-exploitive applications, are writing for Win32, or whatever your plans may be.

By **DAVID E. REICH**

Extending the OS/2 API

Back in May, at the IBM Technical Interchange in New Orleans, IBM announced extensions to the OS/2 API set. They were dubbed "IBM Developer API Extensions". Whatever they're called when they're released, they are simply new OS/2 APIs for use by application developers. These APIs happen to have the same names, parameter lists, and functions as their Win32 counterparts.

WHAT ARE THESE EXTENSIONS?

The developer API extensions are just that. They are new APIs, or programming interfaces, for use by any application developer. IBM is always adding new APIs to the system. Some are newly published interfaces to existing function (the oft-mentioned "document the undocumented interfaces" request), and some are interfaces to new function added to the operating system. These developer extensions are no different.

The new function list, published at the IBM Technical Interchange and found in a White Paper on the IBM Software Developer's home page on the World Wide Web at <http://www.austin.ibm.com/developer>, is a list of over 700 APIs and 300 messages that correspond in interface and function to Win32 APIs and messages with the same names. The idea behind this is to create a level of source code compatibility between applications written for Windows NT or Windows 95 and OS/2.

WHY?

IBM understands that this API/message list will not create complete source code compatibility, and many people have questioned the motivation behind fundamentally supporting Win32 APIs in OS/2. The major concern among developers is, "Isn't this conceding the API 'wars' to Microsoft,

and isn't IBM now letting Microsoft take control of the API?" This question needs to be asked, and the answer, while a religious subject for many, makes sense when viewed in the big picture.

Let's face it. There are many OS/2 applications on the market. There are many more Windows applications. It is a simple matter of economics. With limited resources on the part of developers, the migration is to the larger installed base. For now, that is Windows. With the single-threaded, 16-bit architecture of Windows, APIs aside, there are fundamental architectural issues that made source code compatibility impractical before now. Even if the API were identical between OS/2 and Windows, the "16-bit-ness" of the applications and the single-threaded architecture of Windows would create OS/2 applications that would not exploit the advanced functionality offered by OS/2 and as such, no compelling reason for people to buy them.

Now that applications supporting multithreading and 32-bit function are being developed, it makes much more sense to do something like this now rather than earlier. So why does IBM want to support Win32 APIs?

The plain fact is that there are developers creating Win32 applications. In the past, OS/2 applications have not kept pace with the functionality that was offered in their Windows counterparts, because Windows was being developed for first. What the market was seeing was a MyApp version 5.0 for Windows, while version 4.0 was the current native OS/2 version of MyApp. This could not continue if OS/2 was to thrive and grow.

The primary hurdle is money. (Isn't it always?) Maintaining two sets of source code to do the same thing on two different



David Reich

platforms does not make financial sense. If one could leverage the source code and reduce the cost to keep the same features in both OS/2 and Windows versions of the same application, it would, to a considerable extent, level the playing field to get applications on OS/2, with the same level of function. This is better for both OS/2 and application vendors in that they can sell more.

The Developer API Extensions add new functions to OS/2 (such as common dialogs, DDEML, and Registry functions, among others) and provide interfaces to the existing function so that you can just recompile a significant portion of your application and have native OS/2 functionality. In fact, IBM analyzed over nine million lines of shipping Windows application source code to determine which list of functions to support first, in order to create an approximate 80% common source code base. The other 20% will be operating system-specific function, such as Workplace Shell exploitation, as well as other functions the other platform does not support. Figure 1 shows how you can manage your source code efficiently to make the most use of both environments.

So, in getting back to the question of conceding the API to Microsoft, IBM is tactically supporting the APIs that enable inexpensive migration to OS/2. This provides the buying public with applications that have the same functions on both OS/2 and Windows and lets them choose the platform that best fits their needs. No longer does the application have to dictate the platform, and users who have their operating environment provided for them can use your applications on either.

ADVANTAGES

Some of the advantages for you, the programmer, are reduced cost for multiple platform source code, lower product documentation costs, and more function added to OS/2. After all, if the application is common in features across platforms, why rewrite and republish the documentation?

Another advantage is for customers, because there are lower learning curves (or one curve for customers with multiple operating system installations), and there are products that are up to date functionally. There may be customers who have standardized on an operating platform (or may be contemplating

movement to another), and with your programs, you can be there whether it be Windows or OS/2. The choice is even easier for the customer, knowing your application is just as functional on both systems and with no additional learning for users.

HOW CAN YOU USE THESE EXTENSIONS?

The API extensions are just new functions in the operating system. As stated previously, some of the APIs are just new entry points into existing function, with new names and parameter lists that match what the Win32 APIs use, and some have new function underneath. Examples of new function are the Find and Replace dialog, the Print dialog, and DDEML

see how to structure your code to maximize your investment in source code and leverage two platforms at once.

THE FUTURE

So what's IBM getting at with all of this? Basically, developers should not have to choose which platform to release an application. With a common source code base, you can support different platforms. After all, that is where the software world is heading: reuse and portability.

For your procedural code base, the Developer API Extensions provide that portability. Using a standard C or C++ compiler, you can cross compile the platform independent code (using the extensions) and exploit the appro-

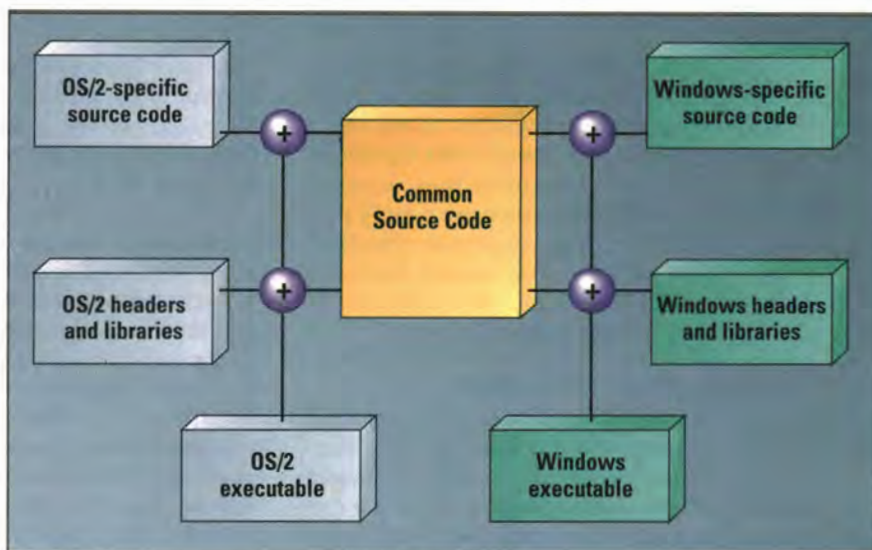


Figure 1. How to manage your source code efficiently.

functions to simplify the use of Dynamic Data Exchange (DDE). Many programmers have requested more common dialogs in OS/2 in addition to the Font and File dialogs already present, and these, with the many new functions, provide a complete and robust programming model.

Using the extensions is easy. Because they are just new APIs, you can use them as you would any other. It's as simple as that. While at the time of this writing, the entry points are not shipping in a publicly available version of OS/2, IBM plans on adding these to future OS/2 versions. You can make use of them in a beta, however, to develop your applications. Even if you don't have a beta, you can simply adhere to the API list in your development and cross compile for the other platform when you're done. Refer back to Figure 1 to

appropriate platform with the other part of your code. This lowers the entry point into other markets where you might not otherwise want to enter, given the costs of a complete rewrite.

If you are already making a move to objects, the IBM Open Class product, along with OpenDoc, will provide you platform independence. IBM Open Class will be on OS/2, Windows, and other platforms, just as OpenDoc is headed there as well. I try to stay away from commenting about things that are not yet shipping, but in the context of mapping out a strategy for you, and "Designing High-Powered OS/2 Warp Applications" (a topic that is near and dear to my heart), while retaining the ability to move to many platforms at a negligible cost, I'll tread softly on that line.

So, for procedural code, the Developer API Extensions get you com-



QA PARTNER: THE NEW GOLD STANDARD IN SOFTWARE TESTING.

THE INTEGRATED QA TESTING PLATFORM — FOR YOUR PLATFORM

"QA Partner's language-based approach offers the best client/server testing solution on the market."

— Hurwitz Consulting Group, Inc.
December 1993

"Honestly, I don't know how I would test this application without QA Partner..."

— Christine Comaford
PC Week, August 15, 1994

"Given our druthers, we would use QA Partner as our one in-house testing system."

— Tim Parker, *UNIX Review*
January 1995

"...Segue's QA Partner. The hands-down winner when it comes to technical prowess."

— Tim Sylvester, *Client Server Today*
February 1995

Segue Software is the world's fastest-growing automated software testing company. No other testing product on the market empowers QA professionals like QA Partner. Call 800-922-3771 for further information on QA Partner today.



Segue

SOFTWARE, INC.

1320 Centre Street
Newton Centre, MA 02159
Internet: info@segue.com

QA Partner is a registered trademark of Segue Software, Inc.

All other system and product names are either trademarks or registered trademarks of their respective companies.

Fast Visual Application Development for OS/2 and DB2



If you're looking for fast and easy application development for OS/2, then take a look at the award-winning Watcom VX•REXX visual development environment. VX•REXX lets you build applications to exploit the graphical user interface, multi-threading, and multi-processing power of OS/2. VX•REXX Client/Server Edition gives you the added power to access DB2 or other database systems, manipulate the

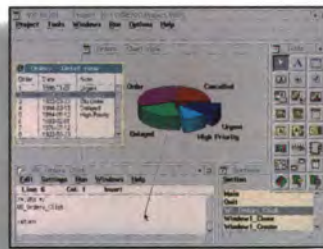
data, and chart the results at lightning speed.

"We like VX•REXX. Using it for development feels like driving a Porsche: it's fast, it's compact, everything's in the right place, and it makes us look good, too."
Peter Coffee, PC WEEK

Designed to Meet Your Needs.

Watcom VX•REXX combines a project management facility, visual designer and an interactive debugger to deliver a highly productive visual development environment. The Client/Server Edition includes additional powerful objects so you can rapidly create rich GUI database applications. You can create OS/2 client applications which connect to DB2/2 or DB2/6000. Use IBM's DRDA support on OS/2 to access DB2 for MVS, DB2/400 for AS/400, and DB2/VSE and VM (SQL/DS) for VM and VSE. Also supported are Watcom SQL and ODBC-enabled databases.

"Overall, this edition of VX•REXX for OS/2 is an outstanding visual client/server development platform." **Nicholas Petreley, InfoWorld**



Point. Click. And Presto!

To create an application you draw user interface objects, customize their properties using standard OS/2 notebooks, and define their event code using powerful drag-and-drop programming. To add database access just draw a query object, visually design a SQL query, press OK and presto— your window is automatically populated with objects that are bound to your query to display, update and search your data.

"Drag-and-drop nirvana." **Nicholas Petreley, InfoWorld**

Give Your Data a Whole New Image.

Energize your applications by displaying your data in a 3D chart. The Client/Server Edition gives you more than a dozen chart types to choose from, along with over 150 display options. You also get complete support for run-time events so you can bring new drama to your data by making your chart interactive.

"VX•REXX is a must buy." **Jacques Surveyer, ComputerWorld**

Standard or Client/Server Edition— Which one is for you?

To start creating powerful OS/2 GUI applications right away, order your copy of Watcom VX•REXX Standard Edition for just...**\$99***

Or, to start creating rich client/server database applications, order Watcom VX•REXX Client/Server Edition for just.....**\$299***

- Over 2 dozen objects, including CUA'91 containers, notebooks, pop-up menus and more
- Integration and control of existing applications through DDE, keystokes or REXX API's
- Easy to learn event-driven programming model with complete on-line documentation
- Support for professional multi-threaded, multi-windowed and drag-and-drop enabled applications
- Code reusability through section and file sharing
- Graphically create CUA'91 Presentation Manager objects, quickly customize their properties, and easily attach REXX procedures
- Package your application as an EXE or PM macro for royalty-free distribution



1-800-265-4555

Watcom
A Powersoft Company

Watcom International 415 Phillip Street, Waterloo, Ontario, Canada N2L 3X2 Tel. (519) 886-3700 Fax (519) 747-4971 *Prices and specifications are subject to change without notice. Price does not include freight and taxes where applicable. Prices quoted in US dollars.
† ODBC drivers are available from INTERSOLV, Inc. Watcom, the Lightning Device, and VX•REXX are trademarks of Watcom International Corporation. Other trademarks are properties of their respective owners. ©Copyright 1994 Watcom International Corporation.

Circle Reader Service Number 7

mon source code. For objects, IBM Open Class and OpenDoc get you to common code. Using these tools, you can spend your time (and money) developing features for your applications in order to differentiate you from your competitors, wow your users, and make yours the applications of choice, rather than worrying about how to maintain function across multiple platforms.

OTHER QUESTIONS

One of the other important questions that I've been asked about the API extensions comes from the concern that IBM is now lowering the entry point to the OS/2 application market for competitors of those who've supported OS/2 from the start.

While this is an absolutely valid concern, several important factors are being overlooked. For example, if you've been developing for OS/2 all along, you've learned all of the tricks, have already fallen into the traps of multithreaded programming, and you know how to debug these types of programs. Developers coming from the DOS/Win16 world have all this to learn. So to start with, your applications will be better threaded on either, or both, platforms.

In fact, if you want to get your OS/2 application onto Win32, you could even do it using these APIs. By moving your existing OS/2 code to the structure outlined in Figure 1, you can get into the Win32 market inexpensively and be the best multithreaded kid on the block.

IBM's goal is to provide you, the developers, with the tools and vision to leverage your investment in code while at the same time drive IBM's business. The Developer API Extensions, combined with IBM Open Class and OpenDoc, will give you the power and portability to make the most of your programming investments. It will also give your users the ability to choose the platform they prefer and choose your applications to run on it.

You have the tools. How you use them is now truly your choice.

David Reich has been with the IBM OS/2 development team since 1987. He has worked on many parts of the system, supported customers and application developers, and traveled the world giving seminars and teaching OS/2. His latest book is, *Designing High Performance OS/2 Warp Applications*, published by John Wiley and Sons. He can be reached on CompuServe at 76711,632 or via the Internet at speedracer@bocarbon.ibm.com.

Take control of your OS/2 scheduling needs with
Advanced Task Scheduler for OS/2

September						
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

ATS

Version 3.0



Are you looking for a way to manage your production job streams? Do you have programs that need to run after the completion of one or more other programs? Do you want to run programs on a regular basis? Do you want to process files that have just been received from another system or modified locally? Do you need to take action if a file is missing? Do you need to schedule around holidays and periods of heavy CPU load?

With ATS you can run your programs when YOU want them to run with out you having to be there.

Upgrades Available
from Version 2 and
other Task Schedulers
Call or E-Mail for Details

MHR

Software and Consulting
2227 U.S. Highway #1 *Suite 146
North Brunswick, NJ 08902

Voice: (908) 821-0359 * Fax: (908) 821-0350 * CompuServe 70312,627

Here are some of the features of
ATS for OS/2:

- * Manage Production Job Streams
- * Programs can be scheduled to run anytime.
- * Programs can be dependent upon Files, Other Programs, Holiday Schedules, and External Signals
- * Integrated Operators Console
- * Logging - To File and Console

New Features:

- * Job Queues - for managing your resources while managing your tasks
- * Periodic Scheduling
- * COMPLETE API and Command Line Interface

Only
\$349

Circle Reader Service Number 8



OnCmd
xBase for OS/2



Take command
with OnCmd®.
The native OS/2®
xBase Database
Development
Environment.

Preserve
your xBase
investment!

- Develop new 32 bit, native PM ready applications
- Migrate existing xBase applications, such as those developed in FoxPro®, Clipper® and dBase®

A Performance
Winner!

- No windows overhead
- Client Server Ready
- Typically index large databases 2X as fast as the competition in 1/2 the disk space
- 350+ Commands/Functions
- Unlimited Runtime License Available




Another fine product from On-Line Data 5 Hill Street, P.O. Box 65, Kitchener, Ontario, Canada N2G 3X4
Phone (519) 579-3930 Fax (519) 579-2130 CompuServe: 70022,104 Internet: oncmd@onlinedata.com

On-Line Data recognizes the following trademarks: OS/2 (IBM Corporation); FoxPro (Microsoft Corporation); dBase (Borland International Inc.); Clipper (Computer Associates International Inc.).

FOR MORE INFO, FAX THE HOTLINE: 519-579-2130

Circle Reader Service Number 9



Migrating an application to a new platform can be a daunting task for a small software company. This time it was worth the gamble.

By CHARLES DIRCKS

OS/2 for the PowerPC: Fact or Fiction

Those of us in the programming community often hear about the promises of future operating systems. If any of us had a dollar for each of those unfulfilled promises, our wealth would be more than acceptable—although still short of Redmond's most famous citizen. Perhaps we should conclude that it is more profitable to make promises than to fulfill them!

Still, businesses are based somewhat on promises of the future balanced against needs of the present. Ours is no different. IBM has promised that the API set would be consistent for OS/2 whether the hardware is an X86 or a PowerPC. Our business needs depend on this promise. Our programming, we felt, would push the promise to its limits: the replacement of the WPObject Class. This article discusses our business requirements, programming approach, and experience with the API set with OS/2 as it applies to both the X86 and PowerPC platforms.

BUSINESS BACKGROUND

Pinnacle Technology was founded three years ago with four primary programming

objectives: make the OS/2 shell "user proof," make OS/2 secure, make OS/2 desktops centrally manageable, and conserve bandwidth.

These goals seem somewhat intimidating to most programmers until you consider that the nucleus of our development team was from American Airlines Sabre: one of the world's largest commercial networks. Using our tools, system administrators can centrally create an ideal (and secure) OS/2 desktop for a group of users. Whenever and wherever they log-on to their LAN, the users get their desktop with the appropriate restrictions (such as no desktop pop-up menus or no copy or delete rights).

With increased mobility in the corporate marketplace, we architected our file size to be very small. For example, the average uncompressed size of a desktop is about 20K bytes. Compressed, we are less than 2K bytes. Therein lies our interest in the PowerPC. Within its architecture is a chip that offers about twice the speed of the Pentium at less than 1 watt—about the same as a hand-held cellular phone.

The goal is to offer the same secure desktop to a hand-held device as to one that is LAN-attached with the same file. The user should not care whether they are using an Intel machine, a PowerPC, or a hand-held device.

PROGRAMMING APPROACH

Our design focused on efficiency. But paramount to our ability to deliver OS/2 desktops to variable machines is a rigorous adherence to OS/2 APIs and a religious avoidance of any direct manipulation of the temperamental .INI files. The .INIs are, by their

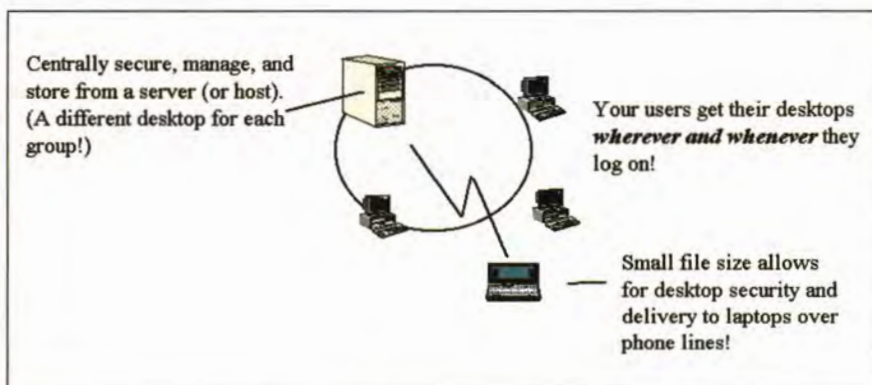


Figure 1. OS/2 desktops are centrally administered and delivered to X86 or PowerPC machines.



very nature, machine- and OS/2-version-specific. This point becomes important when the hardware platform evolves to the PowerPC as well. IBM has documented that the content and format of these files be changed only by the provided APIs.

The nucleus of our desktop delivery is a SOM-based program we call Finder. On an OS/2 2.11 machine, Finder is able to quickly locate and hide all predefined objects so that our other tools can build the appropriate and protected desktop. Finder's class is dependent on our class replacement of WPObject. By replacing the WPObject class, we are able to carry our own instance data, making it possible to extend the object out to the network under tight security controls.

Capabilities on a Warp machine are greatly enhanced. Finder can quickly take a picture of the ideal desktop that an administrator is creating for individuals or groups using OS/2 clients and apply various restrictions to each object the user might access. This again is accomplished by a sophisticated object replacement technique that we apply to the WPObject class. Given its importance to our desktop delivery and security products, the Finder is the portion of our code that we chose to challenge the portability of our OS/2 application to the PowerPC Platform.

Replacing a class is not for the faint hearted, let alone the WPObject class itself. It strikes at the core of the operating system's capabilities. We therefore approached the project with at least some trepidation.

The recompile of the Finder went very smoothly with no changes to any lines of the source code. The Metaware compiler performed flawlessly in building our complex DLLs. Installation was the only setback, which was overcome only by repeated attempts to perform the class replacement.

REFERENCES

"Load and Lock! Delivering a Consistent and Secure OS/2 Desktop Over any LAN," *Personal Systems Magazine*, July, 1995.

"1995 IBM U.S. Technical Interchange," *Presentation Notes CD*.

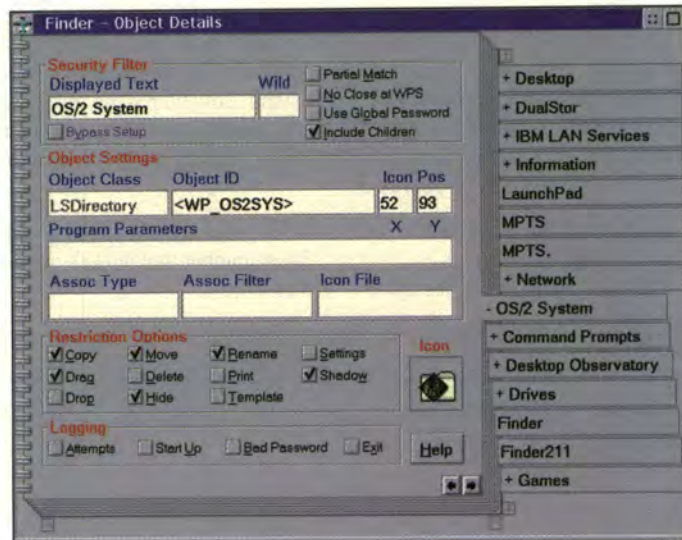


Figure 2. Finder locates all objects on the desktop for administration.

When we filed the problem resolution report, it stated that, "Installation was difficult due to the operating system being in beta." Let's call that a famous understatement.

Once installed, the Finder performed as intended with no apparent differences between the X86 and Power platforms. Our current plan is to provide Finder on the forthcoming Sampler CD for the OS/2 Warp Connect (PowerPC edition).

CONCLUSION

All companies need to conserve resources—some more than others. Still, we were more than curious to evaluate the effort to port our applications to the PowerPC. We asked ourselves what the historical costs were associated with porting existing applications to other platforms (DOS to UNIX, Macintosh to Windows, or VM to CICS). In every case, only the largest companies have been able to bear the development costs of such an effort. Smaller firms do not even make the attempt. Our answer to the question, "Can we handle the cost of porting to the PowerPC?" is a resounding "Yes!"

Charles Dircks is the chief security architect of Pinnacle Technology based in Indianapolis, In. After graduating from Indiana University with a degree in computer science in 1981, Charles has had an extensive career in developing bandwidth-sensitive applications, which included over 10 years at American Airline Sabre. Charles was recently a featured speaker at IBM's Technical Interchange.

Subscribe Today!

As an OS/2 professional, you may be eligible for a FREE one-year subscription to OS/2 Developer, the leading magazine dedicated exclusively to software developers working in the OS/2 environment. This bimonthly magazine is designed for OS/2 application developers who work with everything from LANs to multimedia to DBMS architecture. OS/2 Developer helps you optimize the performance of your OS/2 system and exploit OS/2's complex features to the fullest. It is your definitive source of:

Source code examples
Real-life case studies
New product announcements

OS/2 Programming tips and techniques
Development tool buyer's guides
and more...

OS/2
Developer
THE MAGAZINE FOR ADVANCED SOFTWARE DEVELOPMENT

FAX #: 708-647-5972

To request your FREE subscription simply complete the form below today!

I would like to receive/continue receiving OS/2 Developer free of charge: ☐ YES! ☐ No.

Signature _____ Date _____

(This card MUST be signed and dated, and we MUST have your address, in order to start or continue your subscription.)

Print Name _____

Title _____

Company _____

Address _____ Dept/Mail Stop _____

City _____ State _____ Zip _____

Phone _____ FAX _____

Free Subscription offer good in U.S. only.
For subscriptions outside the U.S. please check and sign below:

- ☐ \$55.95 Canada/Mexico (US funds only)—air mail only
☐ \$69.95 Outside North America (US funds only)—air mail only

Signature _____ Date _____

☐ Charge my credit card (Visa, Mastercard, American Express)

Card No. _____ Exp. Date _____

IMPORTANT:

All questions must be answered in order to start or continue your subscription!

1) My Company is part of the IBM Developer Assistance Program:
1 ☐ Yes. 2 ☐ No.

If "yes", my company's DAP number is _____

2) My job title is (check only one):

- 01 ☐ President, partner, chairman, owner
06 ☐ Vice president, CFO, CIO
07 ☐ Director, head, chief, general manager
08 ☐ Department manager
09 ☐ Supervisor
10 ☐ Other middle management (team leader, project director, etc.)
99 ☐ Other (please specify) _____

2A) My job function is (check only one):

- | | |
|--|---|
| 01 ☐ Software programming | 07 ☐ Computer operations supervision |
| 02 ☐ Application development | 08 ☐ Database Administration |
| 03 ☐ Software engineering | 09 ☐ LAN support |
| 04 ☐ Software quality assurance | 10 ☐ PC software support |
| 05 ☐ Project management | 11 ☐ Engineering, research, sciences |
| 06 ☐ Technical support | 99 ☐ Other (please specify): _____ |

3) What is your company's primary area of business? (check only one):

- | | |
|--|---|
| 20 ☐ Software | 26 ☐ Value-Added Reseller/VAR |
| 21 ☐ Finance/Insurance | 27 ☐ Non-computer manufacturing |
| 22 ☐ Communications | 28 ☐ Defense/Aerospace |
| 23 ☐ Non-defense government | 29 ☐ Wholesale/retail trade |
| 24 ☐ Healthcare/biotechnology | 30 ☐ Education |
| 25 ☐ Computer-related manufacturing | 39 ☐ Other (please specify): _____ |

4) Do you work in IS/DP?

- 1 ☐ Yes. 2 ☐ No.

5) Do you design/write/customize software?

- 1 ☐ Yes. 2 ☐ No.

6) Which operating system do you use, use as a development environment, and/or target for the software you develop? (check all that apply):

Use	Use for Development	Target for Development	Operating System
01 ☐	21 ☐	41 ☐	OS/2 v.2.x
02 ☐	22 ☐	42 ☐	OS/2 v.1.x
03 ☐	23 ☐	43 ☐	Unix (AIX/XENDIX/AUX or any type of Unix)
			MS-DOS/PC-DOS
04 ☐	24 ☐	44 ☐	Windows 3.x
05 ☐	25 ☐	45 ☐	Windows NT
06 ☐	26 ☐	46 ☐	SQL Server
07 ☐	27 ☐	47 ☐	Apple System 7
08 ☐	28 ☐	48 ☐	Other Macintosh OS
09 ☐	29 ☐	49 ☐	OS/2 Warp
10 ☐	30 ☐	50 ☐	OS/2 for Windows
11 ☐	31 ☐	51 ☐	Windows 95
12 ☐	32 ☐	52 ☐	OS/9
13 ☐	33 ☐	53 ☐	Other (please specify) _____
19 ☐	39 ☐	59 ☐	None of the above
20 ☐	40 ☐	60 ☐	

18) Are you developing for, or supporting Lotus Notes?

- 1 ☐ Yes 2 ☐ No

HADT95

FAX this to 708-647-5972
or mail to OS/2 Developer, P.O. Box 1079 Skokie, IL 60076



A young, bleeding-edge software company migrates a bulletin board system to OS/2 for the PowerPC.

By **STEVEN TOWER** and **JOHN MORRIS**

Porting to the PowerPC

Editor's note: AdeptSoft is one of those brave companies that signed up for IBM's OS/2 for PowerPC beta program. Since many of our readers are contemplating the development or migration of applications to this new platform, we asked AdeptSoft's pioneers to share their experience with us.

On the road to IBM's OS/2 for the PowerPC, we expected a bumpy ride with lots of twists and turns, but instead we found the Autobahn. IBM promised that OS/2 for the PowerPC would be a painless move from the OS/2 Intel platform as long as we met one main criteria: the application had to be a pure 32-bit OS/2 application, or using as few 16-bit calls as possible.

LOOKING AHEAD

About three years ago, AdeptXBBS started as a 16-bit OS/2 PM application. In a bold move, we made the decision to rewrite our application from the ground up for the new 32-bit OS/2 with its Workplace Shell. This was a fairly drastic step. One of us had experience ported 16-bit DOS applications to 32-bit OS/2. It was actually easier to do this because the DOS applications didn't have any 16-bit OS/2 API calls in them. Quite a bit of the OS/2 API had changed, either through the use of LONGs instead of SHORTs, or the more intuitive API naming scheme. Not to mention that some API calls had been dropped in favor of improved ones. At the time, the changes needed to make the program fully 32-bit left a bitter taste in our mouths. But it was one of the best moves we made in our development process. Once the conversion was complete, things were once again looking up.

Being relatively familiar with IBM's Workplace OS/Microkernel strategy, and with some guidance from insiders at IBM, we kept on top of the design of Workplace OS/2 (now OS/2 Warp Connect [PowerPC edition]), following every move and making sure that we didn't use any unsupported APIs.

One major decision, as part of our plan

to make sure our application would run well and be able to follow IBM's operating system strategy, was to make sure that potentially nonportable code was put into a .DLL. By placing 16-bit and other nonportable calls into a .DLL, we reduced our porting time extensively.

Over the last three years, development of AdeptXBBS continued at a blinding pace. In December, we received confirmation that we were in the IBM PowerPC beta program and would have our chance to prove that everything we had done was going to work.

OUR FIRST POWERPC COMPILE

IBM made it clear that if we had a pure 32-bit application, porting would be a simple, undaunting job. We still expected problems. The first beta (alpha?) release of the operating system and the accompanying compiler were, shall we say, interesting. We decided to start by porting the hardest part of our product, the two .DLLs, with 16-bit calls. We figured that if we could get those two .DLLs to work, the rest would be a breeze. After a few days, we made our first test run. It didn't work. The operating system returned bizarre return codes that even some of OS/2's developers couldn't explain.

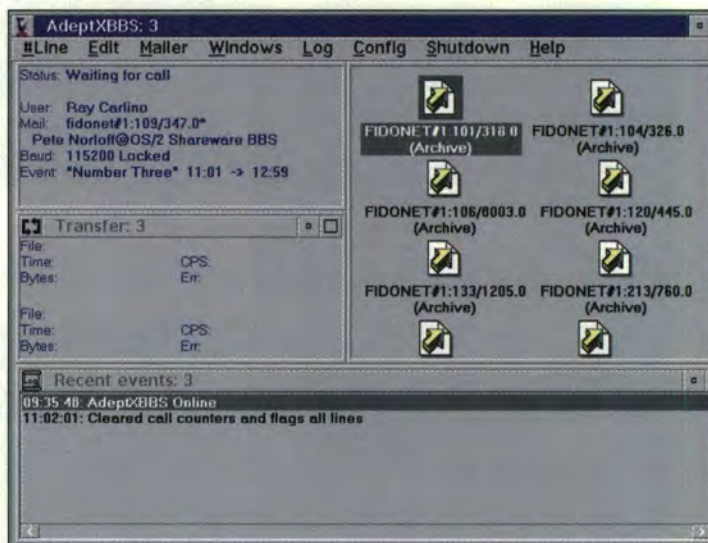


Figure 1. How little things can come back to "haunt" you.

At this point we decided to wait for beta #2 and just clean up the rest of our application code in preparation for the big port. Our application, a 32-bit PM BBS program, was still compiling with several warnings whenever we did a full compile. We decided to make sure our product would compile with absolutely 0 warnings and 0 errors.

FROM WATCOM TO IBM TO METAWARE

For two developers with over 150,000 lines of code sitting in front of them, this was no easy task. We started com-

piling with Watcom's C v. 10.0. Once we had all of the .EXEs and .DLLs compiling with 0 warnings and errors under Watcom's C compiler, our next step was to get the same results with Watcom's C++ compiler. While doing this, several major problems appeared and were corrected. Next, we compiled on IBM's CSet and CSet++ until the same 0 warnings and errors were achieved. We completed the process with Metaware's C++ compiler. We learned that by taking the time to do something that seemed trivial at the start, in the end we had a application

that was not only able to compile flawlessly under three compilers, it was also faster, smaller, used less memory, and was more stable!

WHO MAKES THE HIGHC COMPILER, METAWARE?

The compiler found on the IBM DevCon CD ROM was used for the final porting process. This compiler did not appear to build a functional executable file, but we were able to use it to get an understanding of how the HighC compiler differed from Watcom's C or IBM's CSet++. The first thing that we discovered was that we didn't need to make any changes to our OS/2 API calls (Yeah!!!). Next, we discovered major differences in compiler syntax between the Watcom compiler and the HighC compiler. This is where 95% of the port occurred.

NEW COMPILER SWITCHES

As with most compilers, HighC has its own unique compiler switches. Learning these new switches is the first step to porting your application to the new PowerPC machine.

TYPE CHECKING

HighC has much stricter type checking. Get used to it. You may not like all of the compiler warnings, or errors, but after correcting these problems you will have a more robust application. Figure 1 shows how little things can come back haunt you.

MEMORY ALLOCATION

Make sure you cast all of your memory allocation calls! We had properly cast about 85% of ours, but the last 15% was what got us. Our application is heavily multithreaded, so we make several calls to the well-known `_beginthread()` function. The HighC version differs from the call that most of us have gotten used to in that it has only three parameters. The pointer to a previously allocated stack buffer is not present. This differs from the Watcom and IBM calls, which take four parameters rather than three. The four-parameter calling convention is a leftover from the 16-bit days and was apparently kept the same, so you wouldn't have to change this call if you were changing platforms. Since we have `_beginthread()` calls in several different locations, we decided to just write a macro and put it in one of our header files. That took care of the `_beginthread()` problem.

```
#define _beginthread(w,x,y,z) _be-
```

Single Key-Stroke Screen Printing



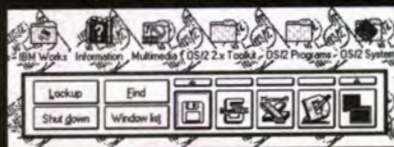
**FOUR
Integrated
Utilities**

- Screen Image Print/Capture
- Clipboard Import/Export
- Screen Saver
- Date & Time Display

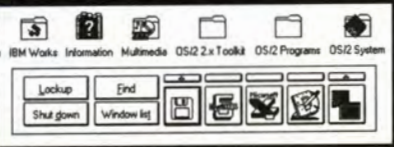


**"The Leader: PrntScr[®]
Screen Image Utility for OS/2"**

- Print/Capture any portion of the Desktop. Including Drop-Down & Pop-Up Menus, and Clipboard Images.
- Print to any Local or LAN attached printer.
- Capture to .WPG, .MET, .BMP, .TIF, .PCX, or .GIF files.
- Quality Color or Black & White copies of color Desktop Images. Don't put up with mush when you can have readable, high-contrast images using the PrntScr "Mapping" feature.

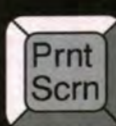


Without PrntScr



With PrntScr

"The Most Stable and Reliable Product Available!"



MITNOR Software
(918) 357-1628 or Contact Your Favorite Retailer.

Circle Reader Service Number 10


```
gintthread(v,y,z)
```

We make heavy use of the variable `_threadid` as an easy way to determine the current thread ID. Watcom C defines this as:

```
extern int __threadid(void);
#define _threadid (__threadid())
```

Our code would say something like:

```
printf("I am thread #%d\n", *_threadid());
```

Under HighC, this failed miserably. It is defined as:

```
extern int _threadid;
```

In this situation, it is best to make a separate function, which contains `#ifdefs` and compiler specific code, like this:

```
int ThreadID(void)
{
#ifdef __HIGHC__
    return(_threadid);
#else
    return(*_threadid);
#endif
}

printf("I am thread #%d\n", ThreadID());
```

Because HighC seems to primarily a C++ compiler, it also is missing the `min()` and `max()` macros. To be safe, make your own `min()` and `max()` functions:

```
#ifndef MAX
#define MAX(a,b) (((a) > (b)) ? (a) : (b))
#endif
#ifndef MIN
#define MIN(a,b) (((a) < (b)) ? (a) : (b))
#endif
```

FINALLY, OS/2 FOR THE POWERPC!

Finally Beta #2 of OS/2 for the PowerPC came out. We were excited. After installing the new build of the operating system we immediately went to work on our video system: a 16-bit .DLL requiring the most work. Since we already had a 32-bit version of this .DLL (that took about a day to convert), it just was a matter of recompiling under the new version of the compiler, writing a test program, and moving the files to the PowerPC. It was a success! The .DLL worked just as well as its 16-bit cousin. With this first success we were energized. We immediately went to work compiling

the other .DLLs directly from the original Intel-based source code. Once again, success after success after success! In about eight hours, we had managed to compile all of our .DLLs and our main .EXE's for the PowerPC. Of course the big question still was, "Would it work?"

Our software makes use of everything that is OS/2. We make extensive use of multithreading and semaphores. We even wrote our own open, read, write, and close functions to bypass the compilers and use nothing but 32-bit calls. We were sure that we

would find something that didn't work.

THE BIG MOMENT

Moving four megabytes of files to the PowerPC was easy. IBM has a basic TCP/IP stack that allowed us to use TFTP to move the files. Now the time had come for us to actually run our software. We typed Adept.Exe at the command prompt. Nothing. The screen blinked, the operating system returned an ambiguous return code, and we were back at the command prompt. Since we were not able to get the de-



Quadron development tools for IBM ARTIC cards:

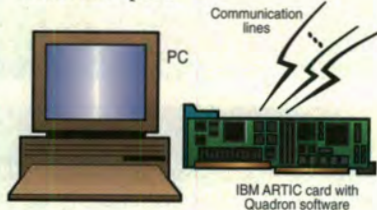


How to communicate better.

Increase system efficiency.

Our development software allows you to shift communication tasks from a PC to an IBM ARTIC co-processor card specially designed to handle them.

You'll gain CPU independence and multiple high-speed ports. Plus, each port on the IBM card can potentially support a different protocol and a different line speed.



Choose just what you need.

We have off-the-shelf solutions for your everyday data transmission needs, and tools for custom situations. You can select from HDLC/SDLC, Bisync, Async, X.25, LAP-B and custom protocols.

Your host PC can be an ISA, EISA, or MCA machine running various operating systems. And possibly most important, you can port co-processor code that you write from one operating system to another with little or no modification.



Serve vertical markets.

Quadron and IBM have teamed up to help people worldwide manage demanding applications such as:

- Travel reservation
- Stock market data delivery
- Telecommunications switching
- Retail point-of-sale purchases
- Process control
- Automatic teller services
- Shop floor control.

Work simpler, work smarter.

Quadron software is easy to install, easy to use, and downright productive. The high-level, C-language API makes co-processor programming simpler. Our user manuals have numerous code examples to speed you on your way. And our support, should you need it, will quickly serve up accurate answers to questions, whether they come in by phone, fax, or through our BBS system.

Act right away.

For today's dependable solution to tomorrow's communications needs, contact us now.



Quadron

209 East Victoria Street
Santa Barbara, CA 93101 USA
fax 805-966-7630
telephone 805-966-6424



© 1995 Quadron Service Corporation. All trademarks are the property of their owners.

bugger to work, we were dead and blind. Resorting to DosBeep, we followed the execution line of the code only to realize that we were missing a .DLL. Digging through our backups, we found the source, recompiled and moved the .DLL over to the PowerPC. Once again, at the command prompt:

[C:\ADEPT\] Adept.Exe [ENTER]

The screen blinked. We assumed, "Oh no, here we go again!" But no. A PM window was drawing on the screen—slowly—but it was drawing. After about 45 seconds, we were staring at our BBS software running on an IBM PowerPC! It worked!

IBM had done it, we had done it!

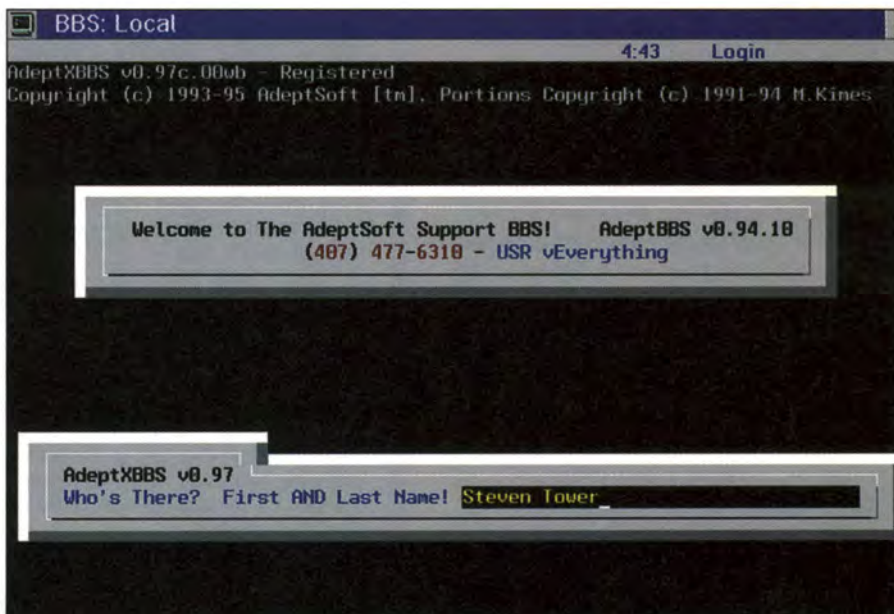
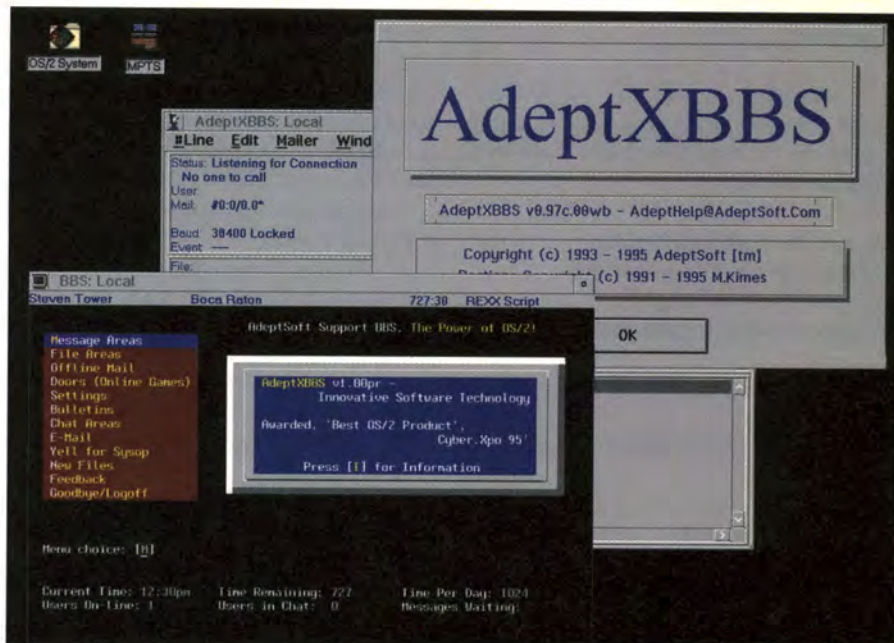
We played with the program to see if everything was working properly. Amazingly, everything worked and ran flawlessly. In just 12 hours we were up and running with no problems on IBM's OS/2 for the PowerPC.

IBM wasn't kidding. When they originally promised that Workplace OS would have the same 32-bit API as OS/2 2.0 we may have all turned around and snickered a little bit, but with OS/2 for the PowerPC they really came through.

Steven Tower is president and cofounder of AdeptSoft, a communications software development company based in Boca Raton, Fla. Tower is the strategic motivation behind the rewrite of XBBS-OS/2 from a 16-bit BBS program into AdeptXBBS, a pure 32-bit BBS program, for OS/2 on both the Intel and PowerPC platforms. AdeptSoft's development team is committed to the creation of innovative communication software that seamlessly integrates all aspects of Fidonet and Internet. Reach him at stower@adeptsoft.com.

John Morris is senior VP/software engineering at AdeptSoft. Morris is codeveloper of AdeptXBBS, a veteran Sysop and expert in communications software. Morris, a successful telecommunications shareware author for nine years, began BBS development by contributing code modifications and utilities for RBBS-PC, the first PC bulletin board software program. Morris was also lead developer of a popular on-line BBS game program, TradeWars 2. Morris has been running the same BBS (The Abandoned Land) for 9 1/2 years, 365 days a year, 24 hours a day, or 83,220 hours total. Contact him at jmorris@adeptsoft.com.

Adeptsoft can be contacted at sales@adeptsoft.com via email or our Support Board: 407-477-6310 or Telnet adeptsoft.com





A veteran OS/2 application developer makes the trek from Intel to PowerPC. His results may surprise you.

By RANDELL S. FLINT

Getting There From Here: Moving to the PowerPC

Now that OS/2 Warp Connect (PowerPC edition) is almost here, and you've heard about all the companies that are moving their applications over to this powerful new hardware platform, you've decided to move yours as well. There's no doubt that you've got an interesting task ahead.

The great news, however, is that you're almost done already, provided you have a (pure) 32-bit native OS/2 application.

Of course the process is much more complex if you have a 16-bit application, or one that isn't OS/2 native. To get to the PowerPC from there, the first (and by far, biggest) step is in getting to 32-bit native mode running on regular old OS/2 on an Intel-based system. But, that's not the focus of this article.

What we're concentrating on is what you can expect in moving that application from the Intel-based OS/2 world over to the new PowerPC-based OS/2 environment. At Sundial Systems, we've been working with OS/2 for PowerPC for over seven months. In that time, we've "moved" virtually all of the code for our Relish® time and information organizer products over to the PowerPC. What we presently have up and running is essentially a complete version of the single-user product, Relish 2.2. It looks, feels, and (for the most part) acts just like the commercial (Intel) version sold today.

We've settled on the term "moved" to describe what we've done for good reason. Bringing a native OS/2 program onto the PowerPC isn't a porting process (like between operating systems) or even a conversion process

(which often describes the process of going from a 16-bit to 32-bit implementation). It's something much simpler! It's very close to a recompile, relink, and go process. And getting closer every day as the operating system and tools reach their final release date.

Things are a little rough today in those places where the OS/2 for PowerPC beta code hasn't completely settled down yet. (This discussion centers around what is known as Beta II from April; a later beta should be available before you read this article.) For instance, in the Beta II version, there's a quirky little bug in the allocation of shared memory, which means two Relish operations—dragging a note to reschedule and double-clicking on a note to revise—fail randomly about 20% of the time.

And there's another weird problem shared by Relish, the System Editor, and a few other applications: the keyboard and mouse button state information messed up whenever the application first gets focus.

There are a few other similar problems and a bunch of little nit-picky ones. But all in all, even with these beta underpinnings, OS/2 for PowerPC has shown itself to share OS/2's heritage of a stable environment on which to build and deploy robust applications. Our latest prerelease of "Power Relish" is at least as stable and snappy as what some companies (mostly on other platforms) might be tempted to sell as a 1.0 product in the retail marketplace.

In the process of getting to this point, we have made *some* changes to the code. Most are workarounds for current problems in a particular build



of OS/2 for PowerPC. A few are permanent changes to the code that replace an Intel-specific construct with one that works equally well on Intel and the PowerPC.

The actual number of code changes is incredibly small, however. Over the course of the past few months, the total number of changed lines of code has been down to as little as five—out of over 250,000! At the moment, that number is a little higher, as a result of working around some Beta II problems to produce a version that is stable and complete enough to distribute to our beta test customers. Even at its highest point, however, the number of changes was almost two orders of magnitude less than the number we needed to go from 16-bit to 32-bit OS/2 several years ago.

To put it in a nutshell, we don't expect to do much more work on Power Relish before it ships. What we will do primarily involves "undoing" some of the things we've done already and performing extensive final testing. That "undoing" really amounts to the removal of the workaround code in the current (prerelease) version once there is a later OS/2 for PowerPC version that fixes some of the problems mentioned above. There's an excellent chance we will be ready to ship *before* OS/2 for PowerPC becomes generally available.

And things are only getting better. Many of the early problems with the development tools have disappeared. Other improvements have been made so that you no longer need to know many of the special compiler flags or idiosyncrasies of the linking process. It now seems quite likely that, had we started today rather than with the prebeta environment of last November, it would have taken only a couple of days, and certainly less than a week, to get Relish that far down the PowerPC path.

The main reason why this whole process is so simple is that we are talking about OS/2. Whether it's running on Intel hardware or PowerPC hardware, the point is that it *is* OS/2. In the future, it will be "OS/2 and more," as the operating system (and the applications running on it) take further advantage of the PowerPC hardware.

Much of IBM's code that implements OS/2 as we know it today is the same on both Intel and the PowerPC. For instance, there are very few changes (and, possibly by now, none at all) in the code that implements the Workplace Shell. The differences, today, are in the "lower" parts of the system: the PowerPC version is based on IBM's microkernel, and sitting on top of that, a new OS/2 Server that implements all of the OS/2 calls (principally, the Dos* calls) in a way that is compatible with the existing Intel implementation.

Eventually, there will be even less of a difference. There will be one version of the entire OS/2 source code tree—one based on the Microkernel and OS/2 Server—that can be targeted toward either Intel or PowerPC hardware when it is built.

Today, applications can already be built that same way—with one set of source code, targeted for either Intel or PowerPC at build time. That's how we are building Relish. If you have trouble believing this, let me relay a story which you may have already heard about elsewhere since it is so remarkable.

The folks at the Desktop Company have an OS/2 product called XFolder. This is an extension to the Workplace Shell that supports all sorts of new arrangement and undo options for desktop folders. It's extremely useful for keeping your icons neatly organized. At some point in May, the company decided that it wanted to see if it could get XFolder up and running on OS/2 for PowerPC—the IBM Technical Interchange was coming up and Desktop *really* wanted to show both versions.

There was one little problem, though. The Desktop Company wasn't part of IBM's Beta program so it didn't have a PowerPC or a copy of OS/2 for PowerPC. It did have, however, the Beta toolkit off the Developer Connection CD (discussed later). The company tried to get XFolder compiled and linked, but ran into (what appeared to be) a compiler problem. So, when the people from Desktop Co. got to the Technical Interchange, they hooked up with the folks from MetaWare (who sup-

plied the compiler) and got the issue resolved. (It turned out to be a mistaken use of a compiler switch.) Once they had a clean compile and link, they brought XFolder up on a PowerPC that MetaWare had at the show.

Now this wasn't a truckload of code (only about 25,000 lines or so), but it wasn't simple code either. What we are talking about is a Workplace Shell replacement class, the corresponding objects, and a fair amount of user interface code to drive the thing. And it worked, for the most part, the first time out! The best part of the story is that the *total* development time involved was only about five hours.

This brings us to a brief discussion regarding how you are going to build your OS/2 for PowerPC application.

Today, the basic strategy for actually building a PowerPC application is to build it on your Intel-based OS/2 system. While at some point in the future you will be able to build it on the PowerPC itself, using your Intel-based system today makes sense. That's because OS/2 is a stable, robust development environment. So, since many developers already use it as a host environment for developing non-OS/2 applications, why not use it to develop PowerPC applications as well?

This Intel-hosted tools approach has the added advantage of allowing development to proceed at a faster pace than would otherwise be possible. And that's because you can develop an OS/2 for PowerPC application *completely* without even having a PowerPC (or PowerPC version of Warp).

While it has gone almost unnoticed by many developers, all the tools necessary to build an OS/2 for PowerPC application have been appearing on IBM's *Developer Connection (DEVCON)* CD since Volume 6 in early 1995. Here's a brief overview of what you'll find:

- A version of MetaWare's High C/C++ Compiler for OS/2. This is essentially the same compiler that you can buy to build OS/2 for Intel applications, but this one has been modified to generate PowerPC object modules rather than Intel object modules. In fact, this is the compiler that



was used by IBM to build much of the Beta versions of OS/2 for PowerPC itself. (For those of you who just can't part with the idea of using IBM's CSet++ compiler, an Intel-hosted PowerPC version is also expected to be available soon.)

- A replacement linker (also supplied by MetaWare) that generates PowerPC executables. These are in a new format specifically designed for the PowerPC.
- Replacements for many of the supporting tools. MultiRes (MRES) replaces RC, MultiLib (MLIB) replaces LIB and IMPLIB, and the Universal Resource Editor replaces the Dialog Editor, for example. In general, these new tools are designed for cross-platform use—they aren't PowerPC- or Intel-specific like the tools they replace.
- A special version of IPMD that allows you to debug your PowerPC application over a TCP/IP-based LAN (but for that, of course, you need a PowerPC and you "need" bugs that need debugging).

In all likelihood, your PowerPC application won't have any bugs! I can hear you saying, "This guy has lost his mind! All new applications have bugs." Yes, it's true that newly programmed code almost inevitably has bugs. But you're not dealing with new code here.

If you have a working, debugged, native, 32-bit application running on OS/2 today, then once you get it recompiled and relinked into a PowerPC executable, it's quite possible you won't have any bugs to debug. And if you do, it's just as likely to be related to some quirk in OS/2 as it is to a quirk in your own code.

What follows are some problems that I've run into during my trip through the process.

DOS* CALLS

Because most of the Dos* calls in OS/2 for PowerPC have been re-implemented by the OS/2 Server and the Microkernel, this is the least "seasoned" part of the system. In a sense, this is the part of the system that will be the most 1.0-like when it ships. It's the most likely part to have a few bugs or to

at least operate a little differently than the present Intel version. It's also possible that some of the parameter validation on these calls will end up being different.

For instance, Relish happens to use the DosQueryCurrentDirectory call in one of those less than common ways. What it does is pass 0 for the size of the buffer that will be filled with the directory name. What OS/2 is supposed to do is return an error code (ERROR_BUFFER_OVERFLOW) together with the size of the necessary buffer. Well, in the initial re-implementation of the Dos*

calls, that didn't work, and the initial testing of the re-implemented APIs didn't catch it. While the problem has since been corrected, it seems likely that a few of the more obscure corners of the Dos* calls may end up working not quite the same until *your* code happens to find the difference.

THE NEW 32-BIT PORTABLE TOOLS

Some of the tools you've commonly used to build your applications are being replaced by new, more portable versions. In the process, the new tools are often



Using Micro Focus or CA-Realia Workbench to Offload Your Mainframe Development?

X:Change™ is what you've been waiting for!

"In X:Change we found a way to overcome a barrier to programmer productivity and to leverage the latent power in the PC development products we purchased."

- Darrell Harper, USAA

X:Change is the fully-graphical TSO replacement that improves programmer productivity and saves mainframe resources:

- ✓ Consumes less than one third the resources of TSO. Transfers files over seven times faster.

- ✓ Full TSO replacement for managing and transferring text and data files, submitting and reviewing batch jobs (JES2&3), even changing security passwords.

- ✓ Single-step, drag and drop text and data file transfer - PDS(E), VSAM, sequential.

- ♦ Record-level data file transfer - sub-set VSAM files on the fly!
- ♦ No REPRO, VRECGEN, WFL or reformatting necessary.
- ♦ Provides direct integration between PAN/LIB and PVCS.
- ♦ SmartCopy only new or changed components, identify dependencies.
- ♦ Scan and synchronize mainframe and PC/LAN libraries.

- ✓ REXX interface and programmable API.

To find out how X:Change makes TSO, IND\$FILE, SDSF/IOF and other MVS utilities things of the past, call today to receive your free copy of "How to Connect" and to arrange a free 30-day trial.

Outside the US and Canada call 415-696-1800
FAX 415-696-1776 • email info@serena.com

800-457-3736



Circle Reader Service Number 11

supporting 32-bit constructs in place of older, 16-bit ones. For example, while resource identifiers used with RC have always been restricted to 16-bit values, MRES does its internal computations in 32-bit form. Thus, if you used tricks with RC (like using -1 instead of 65535 as we did in a few places for historical reasons), you'll need to make changes because the values won't come out the same with MRES.

CODE CHECKS MADE BY THE COMPILER

Whenever you change compilers (or, for that matter, compiler versions), you may also find that your source code gets treated just a bit differently than had been before. Some compilers are tighter about older or nonstandard code structures than others. And some just have different interpretations of what is good and what is not, especially in areas beyond the bounds of the language standard. This isn't a case where one compiler is necessarily better than the other—they are just "different" in their approach to your code.

For instance, the MetaWare compiler caught a couple of places in some of our header files where there was a blank after the \ at the end of a line in what was supposed to be a continuation of a macro definition. What the compiler did to that poor macro definition wasn't a pretty sight—but it was right—the blank shouldn't have been there. CSet++ was just being overly kind to the definition in letting it through.

NON-ANSI C EXTENSIONS

While it's generally desirable to stay within the bounds of ANSI standard C, sometimes there is good reason to stray a bit. Adding threads to your program is a good example. If you try to do that within ANSI-standard C, it means using operating system calls (such as `DosCreateThread`) to control the threads. Once you take this approach, however, you can't use a large number of your favorite C library calls because either the libraries aren't thread-safe or they require some sort of (non-ANSI) setup on a per-thread basis. So, the usual solution is to use nonstan-

dard library calls (such as `_beginthread`) rather than operating system calls for thread control. That's fine so long as the library calls are supported in the same way by the different compilers that you might use. But since, by definition, these calls aren't standard, they might be different, or treated differently, and that's definitely something you need to check.

A perfect example is `_beginthread`. Even though it's part of both the CSet++ and MetaWare libraries, the parameters to the call aren't *quite* the same. The CSet++ version takes four parameters, and MetaWare version takes three. Ironically, the parameters are the same except that CSet++ expects that its extra one (in the second position) be NULL! (For those of you who might not be familiar with it, CSet++ takes this extra parameter for historical reasons, which are related to how threads were handled on OS/2 1.x.)

THE TOOLS ARE STILL EVOLVING

Most compilers don't include all the non-ANSI part of their header

DevTech
Proudly
Introduces...

DeskMan/2™

The Award-Winning, Multi-Function Desktop Enhancements for OS/2®

Productivity Pack

The DeskMan/2 Productivity Pack...Gives you 5 powerful applications and 9 great deals in one box for one low price!

DeskMan/2 v.1.51b...THE premier solution to manage and enhance the OS/2 environment. This best selling package allows users to install, configure, backup, secure and manage desktops and objects locally and remotely. The **VUEMan/2** component brings new meaning to "multi-tasking", enabling users to create as many as 81 virtual desktops. A simple drag-and-drop, point-and-click interface makes **DeskMan/2** easy to use for novice OS/2 users, while its API and full CID enablement provide powerful command and control of the OS/2 operating environment for corporate administrators. **DeskMan/2 v.1.51b** includes features that were previously available only in the Corporate Edition such as password protection of objects, and Workplace Shell audit trails.

FREE* Upgrade to DeskMan/2 v.2.0 when available!

*plus shipping and handling

PLUS

DCF/2 Lite...The "on the fly" data compression package for OS/2. Use the DCF/2 to create multiple, HPFS-formatted virtual disks on top of your existing FAT or HPFS formatted partitions, network drives—even floppies! DCF/2 is on-line backup without an expensive tape unit or individually archived files.

The Graham Utilities...The most comprehensive suite of disk, file and general utilities specially designed for OS/2. Combined with networking options, HPFS utilities and system diagnostics, The Graham Utilities provide all OS/2 users with invaluable support. This Limited Edition specifically created for the Productivity Pack offers 15 powerful functions of the 52 invaluable applications and modules that are available in the full version.

Relish v.2.12...Winner of 1994 OS/2 Magazine's Editor's Choice Award! Flexible scheduling functions combine with the Workplace Shell drag-and-drop environment for ease and convenience. Relish's intuitive approach to personal calendaring is grounded in a multi-tasking, multi-threaded design. Combine that with drag-and-drop enhancements, time-saving features, and extensive flexibility to manage time and tasks with ease.

CPU Monitor Plus, Special Edition...The powerful tool to manage your OS/2 computer! CPU Monitor Plus is your complete package of advanced performance and analysis tools, all integrated into one powerful solution. Use CPU Monitor Plus to continuously monitor and display CPU, RAM, Disk, COM port activity, as well as dynamically monitor and control your OS/2 programs! Let the powerful analysis engine rip through raw system data and give you graphs or text, percentages, ratios, bytes or seconds.

PLUS Upgrade coupons for each of the above power apps are enclosed!

*Special editions developed exclusively for the Productivity Pack.

EDITORS' CHOICE
AWARD
OS/2
MAGAZINE
1994

Best of OS/2
1994
OS/2
MAGAZINE
Editor's Choice Award

OS/2
WARP
Ready for
LAN Server

"DeskMan/2 dramatically improves the ability of corporations and users to get the most out of OS/2. DeskMan/2 helps administrators restrict and control what users are doing and provides robust backup/restore features and seamlessly enhances the features of the WPS."
Dr. Mike Kogan: Lead Architect of OS/2 2.0
Author: The Design of OS/2

DevTech
Development Technologies, Inc.
Software Development & Technology Transfer
308 Springfield Road, Forest Acres, SC 29206-2113
Voice: (803) 790-9230 • Fax: (803) 738-0218

Visit Your Local
Software Reseller

Copyright 1995 Development Technologies, Inc. All rights reserved. DeskMan, DeskMan/2, VUEMan, VUEMan/2 and DM2Image are trademarks of Development Technologies, Inc. Workplace Shell is a trademark of International Business Machines Corporation. IBM and OS/2 are registered trademarks of International Business Machines Corporation. Relish is a registered trademark of Sundial Systems Corporation. DCF/2 Lite is a trademark of Proportional Software. The Graham Utilities is a trademark of WarpSpeed Computers, Inc. CPU Monitor Plus is copyright by BonAmi Software Corporation.

files, nor do they automatically include non-ANSI libraries when used for linking. Usually, you include something in the compile options to tell the compiler that's what you want to do. While the MetaWare compiler includes many non-ANSI calls by default, that's not true for thread-related calls like `_beginthread`.

In the early versions of the MetaWare PowerPC compiler, you did this by defining `__MULTI__` on the command line and changing your link options to reference the multithreaded library (`hct.lib`) rather than the normal one (`hc.lib`). By the time Beta II arrived, this had been simplified through the use of the `-thread` compile option, which took care of both the header file and library issues.

AVOID MAKING ASSUMPTIONS

There are other non-ANSI issues that need to be dealt with in a similar way, one of which is SOM support. If you happen to use this in your code, you might think you should use the `-som` option in the same way you use the `-thread` option. Well, that's not quite right,

because when *compiling*, this turns on MetaWare's support for Direct-To-SOM code generation, which is probably *not* what you want in this case. However, you probably do want the `-som` option when *linking*, since this will cause any necessary SOM libraries to be linked in.

LIBRARIES YOU NEED MAY NOT BE AVAILABLE

One particularly vexing problem in these early stages of movement to the PowerPC is the availability of any third-party libraries that you might need as part of your application.

The good news is that libraries move over just as easily as applications do. The bad news is that you need the source code to do it (or you have to have the owners of the source code do it). In Relish, we do use one third-party library. Fortunately, we do license the source code and have moved it over using the same process as our own source.

SOME DIFFERENCES IN THE UNDERLYING IMPLEMENTATIONS

And, finally, once you do get the

code compiled, linked, and running on the PowerPC you may still notice some minor differences that shouldn't (but might) make a difference. That's because parts of the operating system are indeed different.

In Relish, we happened to notice this in the case of critical sections. While it's probably not noticeable to the user, we discovered that entering and exiting critical sections is a little more costly on the PowerPC than it is on the Intel version of OS/2. Sources tell us that it's true, and the reason has to do with how the OS/2 Server interfaces with the Microkernel to implement (jointly) the `DosEnterCritSec` and `DosExitCritSec` calls. Likewise, there are other calls which seem a little faster on the PowerPC, though this may have as much to do with the processor as anything else.

So if you've got any really tight timing dependencies, particularly among multiple threads, you might want to test that code a little extra carefully.

Another area of major differences is in the device drivers. IBM



"THERE JUST AREN'T ANY OS/2® APPLICATIONS"

(YEAH, AND THOSE CRAZY HORSELESS CARRIAGES WILL NEVER CATCH ON)

Hyperwise
by IBM

\$195



Hyperwise V2.0 for OS/2 Warp is a productivity tool for application developers which enables WYSIWYG authoring of hypertext online information and application help. Using Hyperwise, you can create multimedia-rich online books and help text, or effective tutorials and training materials. Key features of Hyperwise 2.0 include: drag-and-drop techniques; link text, graphics, audio and video; JPF bookmaster and RTF import; INF, HLP and HTML export.

30H1731 MSRP\$295.00 Save\$100

VisualAge C++
by IBM

NEW!



VisualAge C++ Version 3.0 is a powerful C++ application development environment that combines visual programming with robust professional tools. Developers can visually build applications from parts and combine the parts to construct programs. This product includes: IBM Open Class Library and an extensive set of programmer tools - C/C++ compiler, editor, browser, debugger and performance trace analyzer.

Full Packs, Upgrades and Additional Licences Available - Call for Pricing!

dbfREXX & dbfLIB

by dSoft Development

New! Version 2

Includes Support For MDX & NTX Files!

dbfREXX:

This dynamic link library provides simple, affordable database management from your OS/2 REXX programs. Now accessing dBase files is unimaginably easy.

DSF25 MSRP \$99.00 IB \$95.00

dbfLIB:

The solution for harried MIS developers! Allows multiple users to access dBase files. Provides a consistent set of API's across OS/2, DOS, Windows & Windows NT.

DSF22 MSRP \$195.00 IB \$159.00

HUNDREDS OF OS/2 PRODUCTS AVAILABLE
CALL FOR NEW FULL COLOR CATALOG

CORPORATE CUSTOMERS - ASK ABOUT ADVANTAGE PROGRAM

ORDERS: 1-800-776-8284

<http://www.indelible-blue.com/ib>



OS/2, Hyperwise, VisualAge C++ and IBM are registered trademarks of IBM Corporation. All other trademarks belong to their respective owners. Prices are subject to change. This ad composed on a PC running on OS/2 Warp, using native OS/2, DOS and Windows apps.



hasn't publicly said so, but it appears that much (if not all) of the device driver code is brand new in the PowerPC version. In part, that's to be expected, since the low-level hardware architecture of the systems is indeed different. While the various OS/2 APIs hide most of these issues, some of the differences will no doubt surface through them. In particular, applications that deal intimately with devices (such as communications programs) would appear to require particular scrutiny in their use of these APIs.

CONCLUSION

All in all, it's fairly easy to move your application onto OS/2 for PowerPC. Start with your 32-bit native OS/2 application, keeping in mind that OS/2 for PowerPC and the tools you need are still *beta* in nature; and, don't be surprised by the inevitable quirks that might put bumps in your path. Then, head out on this powerful new hardware platform. You'll likely enjoy this trip much more than porting or conversion efforts you've been on in the past.

And, if you'd like to get a feel

of just how close things can get, we'll be happy to send you Working Model (demo version) copies of Relish for both Intel and PowerPC. That way you can really compare and see for yourself.

You can download both versions from a variety of sources (including the Sundial Systems support section in the OS2AVEN forum on CompuServe). Or, if you prefer, we'll send you copies on diskette. For the latest availability information, just drop us a note via the Internet to relish@ibm.net.

Randell S. Flint is president of Sundial Systems Corp. As the chief architect of Relish, he's been deeply involved with OS/2 since Version 1.0. His earlier article on the client-server architecture of Relish appeared in the Spring 1991 issue of *OS/2 Developer*. He can be reached via the Internet at rflint@ibm.net or at Sundial Systems Corp., 909 Electric Ave., Suite 204, Seal Beach, CA 90740.

Now, *this* is a genuine
Rocky Mountain "Hi!"



**Calling You To -
ColoradOS/2 1995**

**October 15-
October 20, 1995**

**Keystone Conference
Center
Keystone, Colorado**

Visit our Web page:
<http://www.colos2.com>

ColoradOS/2 is a production of
Kovsky Conference Productions, Inc.

**OS/2 Programming
Summit!**

ColoradOS/2

In the fall, the bugling of the elk can be heard echoing through the Colorado high country. Trumpeting both challenge and accomplishment, these magnificent animals gather every fall to form groups for the year ahead. Solitary for much of the year, and social for the rest, they rely on each other without compromising their maverick nature.

Programmers committed to OS/2 gather in the Colorado high country at the same time of year, for similar reasons: to gain strength and wisdom from their peers, while drawing inspiration from the ever-increasing numbers of those peers.

ColoradOS/2 is renowned throughout the world for the quality of its technical content, but that is only part of its appeal. The rest is the friends and contacts you make here. Isn't it time you were part of this unforgettable experience?

**Plan now to
attend the Fourth
International
ColoradOS/2
!!!**

To register or for information call:
(800) 481-3389 US & Canada
(719) 481-3389 International
(719) 481-8069 FAX

OS/2 is a registered trademark of International Business Machines Corporation.

Circle Reader Service Number 14

REFERENCES

"A Diary of a Programmer's Vacation in Florida," *IBM Developer Connection News*, Volume 6.

"Countdown to the PowerPC," *IBM Developer Connection News*, Volume 6.

"Delivering the OS/2 API Commitment," *IBM Developer Connection News*, Volume 6.

"Development Tools for OS/2 for the PowerPC," *IBM Developer Connection News*, Volume 7.

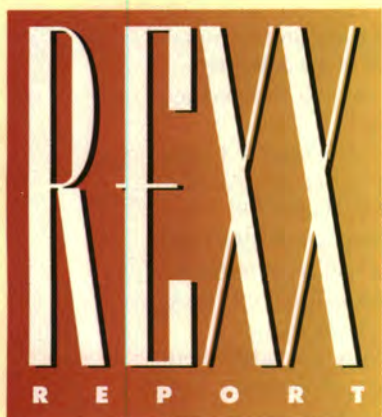
"GRADD Architecture for OS/2 for the PowerPC," *IBM Developer Connection News*, Volume 7.

"OS/2 for the PowerPC: The Development Environment," *IBM Developer Connection News*, Volume 6.

"OS/2 for the PowerPC: Development Tools Coming Online Fast!" *IBM Developer Connection News*, Volume 6.

"Power Applications," *OS/2 Developer*, Volume 7, No. 2, March/April 1995.

Making the Most of REXX



Quantities are limited. Order now!

For fastest service, fax the completed order form below to (708) 647-5972.

Call (800) WANT OS/2 for immediate service. Outside the U.S., call 1 (708) 647-5960.

A special issue by the publishers of OS/2 Developer

Easy-to-learn. . . fast. . . interactive. . . whether you're considering using REXX for the first time—or a REXX developer who wants to maximize this IBM-endorsed language—you'll find the tips and techniques you need in the *REXX Report*.

The 2nd edition of the *REXX Report* has all newly commissioned articles by REXX-perts in the field. For only \$7.95, you'll get:

- A Comparison of REXX Visual Builders

- E-Mail Enabling REXX
- Introduction to Workplace Shell Programming Using REXX
- Combining REXX with PostScript
- PDS Transfers Using EHLLAPI in REXX
- and more, including a Buyer's Guide to REXX language products, interpreters, debuggers, screen generators, and other tools.

Y

es, please send me _____ copies of the **REXX Report!** Each copy is only

\$7.95 in the U.S. or \$8.95 for Canadian or foreign. Please add \$2.50 for shipping and handling.

☐ My check is enclosed. ☐ Visa ☐ MasterCard ☐ American Express

Credit Card Number _____ Expiration Date _____

Signature _____

Name _____

Title _____

Company _____

Address _____

City/State/Zip or Postal Code _____



Mail to *REXX Report*, P.O. Box 1079, Skokie, IL 60076-8079.



An OS/2 software developer meets airline industry requirements with a low-cost, client-/server-based reservation system.

By JIM MONROE

Building an OS/2-Based Airline Reservation System

Since the advent of the PC, software designers have been faced with many choices: operating systems, database engines, programming languages, and other application requirements. This article is about the approach taken and some decisions made in the development of an all-new, client-/server-based airline reservation system. This development approach may provide some help when, and if, you are faced with a similar predicament.

BACKGROUND

Prior to embarkation into the airline industry, many months were spent analyzing the industry and the application offerings available. The findings were that the industry is extremely narrow and, to survive, any products offered would have to be inexpensive and use the best technology available.

SETTING SOME GOALS

An airline reservation system was produced that was scalable to serve the needs of a very small budget airline all the way up to the size of currently available large reservation systems. The system would be priced 50% lower than competitive systems and would still earn a profit.

The design criteria consisted of the following elements:

- It would be scalable from a single PC to an IBM 370/390 system
- It would use an "Open Architecture"
- All data would be available from a relational database
- The operating system would support multitasking and multiple users
- A response time of less than two seconds would be maintained at all times
- Networking would be easy to install, maintain, use, and include the ability to connect via the Internet

- It would be accessible from any terminal type, internal or external to the enterprise
- The system would be easy to use and require little or no training.

A descriptive name was selected that would also convey simplicity: The Airline Reservation System (TARS). With TARS, several potential customer groups are addressed:

- Start-up airlines with a choice of terminal types and physical placement because they have nothing installed—no heritage
- Airlines that have already migrated their reservation center, airport counters, and agents to PCs because they have been migrating to LAN-based distributed processing
- Airlines that have not migrated (or have been delayed) to a PC "terminal" strategy because their installed base or wiring impaired migration
- Airlines that are "renting" services, which preclude flexibility or ultimate use of native PC technology because someone else is dictating the connectivity alternatives
- Airlines with significant investment into wiring that cannot be easily accommodated into certain terminal connectivity because they have stayed with the early VT terminal clusters.

SEEKING A MULTI-USER, MULTITASKING, MULTI-PLATFORM SOLUTION

To provide needed scalability, the architecture was designed as a distributed application within a client/server environment. This would allow the relational database to reside on either the server or on a client system. After reviewing several database engines that met this requirement, IBM's DB2 became the logical choice. One reason was because this engine could be used on OS/2, AIX/UNIX, OS/400, MVS, VSE, VM, Sun Solaris, or HP-UX. This covered all possible system combi-



nations, allowing TARS to be operating system- and hardware-independent.

C was chosen as the programming language because it also supported all operating system and hardware platforms. Also, C programmers are available and the language is not manufacturer dependent.

OS/2 WARP—ON SERVERS AND CLIENTS

At the low- to mid-range level, OS/2 Warp was used for both server and client. By installing DB/2 on this platform, the smallest configuration fits on a single personal computer. By propagating DB/2 to larger or multiple-user platforms, and keeping the client portion on OS/2, the system can grow to any size needed. This also included the ability to attach any nonintelligent terminal directly via RS232, over TCP/IP or the Internet.

When examining various networking requirements, it was found that the IBM LAN support (LAN Server version 2.X and later) addressed the traditional client/server requirements, as well as wide-area networks (WAN). It also allowed nonintelligent terminals to be run off the servers, eliminating the need for some dedicated PC clients. The ability to have clients either local or remote to the server was possible. Complete interoperability of the system was maintained by the DB2 client/server support.

ADDING TERMINAL SUPPORT TO OS/2 WARP

The last portion of the architecture was the terminal server, which connected the central database clients to the airports. The airports provided reservation terminals at the check-in counters and departure gates. As such, the imposing routers and gateways would make the solution expensive and management-intensive. A multi-user system eliminated this problem because the multi-user cluster could be located either at the Reservation Center or at the airport. In addition, remote printing of tickets and reports would be available at both the check-in and departure gates.

The decision was made to use a currently available product, REMOTE-OS, instead of

writing the terminal support in-house. This decision was based on REMOTE-OS's ability to provide both the terminal support as well as the required network connectivity.

PERFORMANCE TUNING

Stored procedures were used to reduce the interaction between the client and server. This alone resulted in an approximate 50% reduction in communication between the two. To improve performance, only static SQL statements were used (as opposed to dynamically executed statements). Sorting and ordering were held to a minimum to reduce execution time. (When DB2 resides on the OS/2 system, as much real storage as possible is made available.) These steps resulted in a system that provided a predictable performance that was within our design criteria.

BUILDING THE FRONT END

TARS provides a friendly user interface including action bars, pull-downs, message windows, and parallel dialogues such as on-line contextual help (Figure 1). The operator interface is self-prompting, self-educating, and provides examples of allowable data for entry. Each operation consistently follows the same steps each time. OS/2 Warp's Presentation Manager was not used since that would limit remote terminals to intelligent PC systems and drive up our costs.

DB2 provides the ability to query the



Figure 1. TARS can provide an affordable alternative for a reservation system.



GET DEBUGGED!

REXX Diagnostic Commander Makes REXX Debugging Easy

Software developers, administrators, and all OS/2 professionals can efficiently code and test REXX command files in a completely interactive environment. Easy to use, RDC enhances your code writing ability while increasing productivity. *Programs are debugged easily and quickly!*

- Supports Multiple Platforms
- Source Code Display
- Variable Display Service
- Variable Alteration
- Single Step Review
- Dynamic Watch Windows
- Logic Alteration
- Deferred Debug Option

**Get the REXX
Diagnostic Commander**

Only \$45⁰⁰

Order It Today!

**Call
800-724-7011**



**Suitable
Alternatives**

4655 Old Ironsides Drive, Suite 220
Santa Clara, CA 95054 • 408-727-3142



Circle Reader Service Number 15

database via menu-driven forms, specifying data fields, logical relationships, and sequences. An example of a fixed "query" is a passenger who calls with questions about a reservation but doesn't remember the confirmation number. The operator can enter the passenger's last name, or a smaller subset of the name, on the panel. The result is a view of similar names with home and office phone numbers, where the operator, prompted by the passenger, scrolls to the correct reservation. At this point, the system displays the outbound and inbound flights for the passenger, and the selection continues until the passenger is fully serviced.

When a graphical interface to DB2 becomes a requirement—because the terminals are GUI capable—the GUI-WYSIWYG look and feel can be added using IBM's GUI DB2 tools, without changing any code in TARS itself. In addition, DB2 provides an export/import function to most spreadsheet or word-processing applications, so the customer is still not obligated to "our way of thinking." Further, DB/2 provides the necessary data security and data integrity support, an absolute requirement for an airline reservation system.

A CLIENT/SERVER AIRLINES RESERVATION SYSTEM

The TARS project achieved the goal of providing the airlines with an affordable alternative for a reservation system, while still providing the function of larger systems. The self-training user interface lets an airline hire agents unfamiliar with airline industry jargon and spend less than a day training them. This alone more than justifies the system in the eyes of airline management, saving several weeks for each counter or reservation person.

The DB/2 database provides comprehensive reporting on a real-time basis. This allows management to make decisions immediately, rather than waiting until tomorrow for a report. TARS is completely configurable, on a real-time basis, and flights can be created, edited, and rearranged as needed. This would be impossible without a relational database.

TARS can be connected with other reservation systems used by independent travel agents. Reservations are simply downloaded into TARS once the local PC has polled the other reservation system(s) and extracted all reserva-

tions made by the independent agents.

TARS has been installed at several airlines and has proven to be successful and the design criteria realized.

The growth of OS/2 Warp, DB/2, and C functionality should pave the way for future enhancements that are not previously available to small airlines, such as adding voice response capability to TARS. From an application development point of view, software designers are positioned to take advantage of hardware and software progress without having to wait.

The nicest part about this project was that once the tools were selected, the focus went to application development. OS/2 provided the multitasking, multithreading environment. LAN Server provided the peer-to-peer and PC support. DB2/2 provided the engine and support for the structured and unstructured queries for an endless variety of reports. Finally, REMOTE-OS provided the local and remote access to PCs and terminals, eliminating a lot of low-level programming.

Jim Monroe was employed by IBM for 30 years until 1989. Since then, he has developed various OS/2- and DB2/2-based applications for the retail and airline industries. He can be reached at 407-361-0109, fax 407-361-9851 or via CompuServe at 74230,2407.

REFERENCES

TARS specification document.

REMOTE-OS users documentation.

DB2 Information & Planning Guide, S62G-3662.

User's Guide to OS/2 Warp, 83G8300.

OS/2 WARP, DB2, LAN Server, AIX, OS/400, MVS, VSE, and VM are available from IBM.

HP-UX is available from Hewlett Packard.

Sun Systems is available from Solaris.

TARS is available from TruData Inc.

REMOTE-OS is available from the The Software Lifeline Inc.



Back Issues

To order your back issues of OS/2 Developer, call now:

800-WANT-OS2
(800-926-8672)

or write to:

OS/2 Developer, P. O. Box 1079
Skokie, IL 60076-8079

Issues are \$9.95 plus \$2.95 shipping and handling.
All orders must be prepaid.

Hurry!
Supplies Limited!



OpenDoc—More Than Just Documents

The forthcoming OpenDoc technology, jointly developed by IBM, Apple, and WordPerfect, is currently promoted as an open-architected compound document technology in current literature. This description provides ample ammunition for authors to compare and contrast OpenDoc and its chief competitor, Microsoft's OLE (Object Linking and Embedding). This viewpoint has been promoted to provide developers and users with a clear product identity against the competition. Currently, much ink is being spilled in comparing and contrasting OpenDoc and OLE as compound document editors.

This is unfortunate because OpenDoc has a deeper and more subtle technical foundation than the description "compound document editor" implies. What OpenDoc really represents to software developers is a plat-

form-independent, language-independent application framework.

Much of OpenDoc's capabilities are directly attributed to the use of IBM's System Object Model (SOM) in implementing OpenDoc. After some early growing pains, SOM has now matured enough to be ready for prime time. SOM is a proven, reliable technology, as demonstrated by OS/2's Workplace Shell. After lurking under the hood of OS/2 for two years, SOM is ready for the spotlight as OpenDoc is unveiled.

The purpose of this article is to present a view of OpenDoc as a prebuilt multiplatform application framework for developers. This viewpoint avoids the abstract document-container-part terminology usually associated with OpenDoc in favor of framework-class-subclass terminology. Framework-class-subclass terminology is used in formal object-

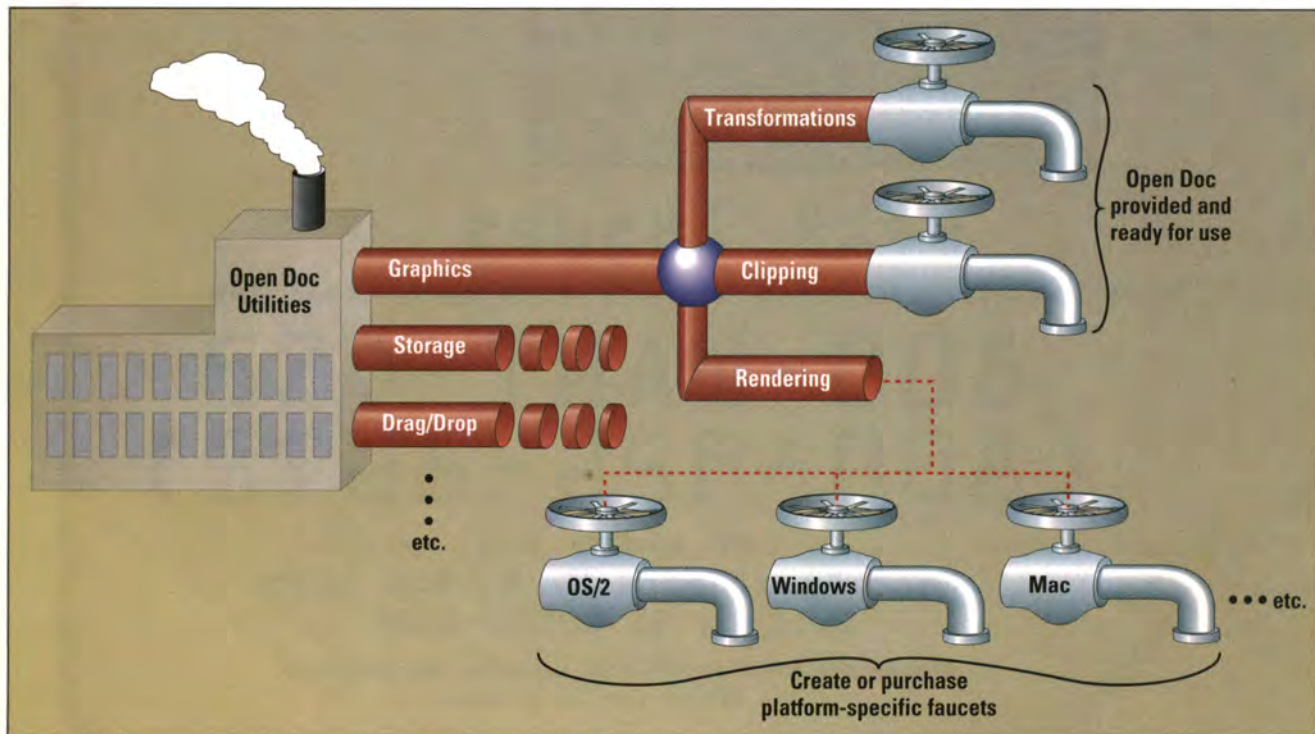


Figure 1. The OpenDoc plumbing system.



oriented design methodologies such as the Booch method. Hopefully, this will allow you to break the association of OpenDoc with documents and concentrate on OpenDoc as an object-oriented development platform, which may be used to develop any type of application.

We will start off by first examining early frameworks and the limitations in their use. I'll then introduce SOM technology, showing how it alleviates the prior shortcomings of frameworks. A general overview of the OpenDoc framework design will liken the OpenDoc framework to a plumbing system complete with pipes, faucets, and caps. Finally, a short OpenDoc sample subclass will be developed to demonstrate the ease of writing a new OpenDoc application. The sample will port the DIVE (Direct Interface Video Extensions) sample program from the OS/2 Multimedia Toolkit to OpenDoc. So prepare to DIVE into OpenDoc!

VANILLA FRAMEWORKS

Frameworks are best described as cooperating object-oriented classes, which when used collectively provide a working piece of functionality. For example, the IBM C++ User Interface classes form a framework that collectively provides an object-oriented programming interface to access the OS/2 Presentation Manager APIs. Many types of frameworks currently exist to solve common problems. A Collection framework provides access to arrays, linked lists, and queue structures, while Persistence frameworks allow objects to save their current state to permanent storage (that is, save data to disk).

Frameworks take advantage of previously written and debugged code and allow framework users easy access to that working functionality. As an added bonus, framework users are able to extend the framework's functionality by subclassing framework classes. Subclassing a framework class allows the developer to inherit all the functionality of the base class and add new functionality, as well as remove or replace existing functionality.

The ability to add, remove, or correct framework functionality makes them very flexible in solving development problems. Adding functionality to a framework is critical to promoting code reuse. Removing existing framework functionality is a prerequisite

to replacing functionality so as to correct framework bugs or to optimize performance.

The power of frameworks to date has been limited to providing developers with existing functionality wrapped externally with object-oriented classes. User Interface frameworks, Collection frameworks, and Persistence frameworks functionally provide

```
#ifndef _DIVEPART_
#define _DIVEPART_

/* Include parent's definition */
#include <SimplPrt.idl>

/* DIVEPart is descended from SimplePart */
interface DIVEPart : SimplePart
{
    /* This new method will update the visible facets onscreen */
    void UpdateFacets(in ODFacet facet);

    /* SOM-specific section */
#ifdef __SOMIDL__
    implementation
    {
        /* Define a prefix for methods */
        functionprefix = DIVEPart;

        /* Define the release order to preserve backwards binary compatibility */
        releaseorder: UpdateFacets;

        /* Select methods to override from the ancestor classes */
        override:
            somInit,
            somUninit,
            FacetAdded,
            FacetRemoved,
            GeometryChanged,
            HandleEvent,
            InitPart,
            InitPartFromStorage,
            ActivateFrame,
            FocusAcquired,
            FocusLost,
            ActivatingWindow,
            Draw;

        /* We will need an ordered collection to contain all visible facets */
        passthru C_xih = "class ODXOrderedCollection";

        /* Define an instance variable */
        ODXOrderedCollection *Facets;
    };
#endif
};
#endif
```

Listing 1. A new IDL source file that describes a new class.



nothing new. Their power lies in making functionality easy to use and to extend. By themselves, frameworks haven't made the chore of writing an entire application much easier.

Frameworks also have had severe limitations in their use. In the past, frameworks could only be used from the programming language, which used to develop the framework itself. A framework written in C++ was useless to a Smalltalk developer and vice versa. Also, different language compilers were even incompatible, most notably with C++. This incompatibility of frameworks isolated them from the increasingly heterogeneous client-server solutions of today and was a limitation on promoting business-wide code reuse.

Another limitation in using frameworks occurred when the framework needed to be updated or corrected. Updating the framework would require all clients using the framework to be recompiled because the framework didn't preserve backwards binary compatibility. This synchronization dilemma made frameworks cumbersome to maintain and update.

SOM FRAMEWORKS

SOM technology breaks down the earlier barriers to object-oriented frameworks. SOM is not a programming language but rather an object packaging technology. SOM is much more comprehensive than the short discussion I will present here, which will serve only to demonstrate a portion of its capabilities in correcting the prior shortcomings of frameworks.

SOM allows frameworks to be used by any programming language that has SOM bindings available. Currently, SOM bindings exist for C, C++, Smalltalk, and ObjectREXX. COBOL bindings have been indicated to be forthcoming shortly. As you can see, even procedural (nonobject-oriented) languages such as C, COBOL, and REXX can use frameworks using SOM technology.

SOM also preserves backwards binary compatibility, so that when a framework is updated the client programs don't have to be recompiled. Backwards binary compatibility makes it feasible to package frameworks as shareable binary DLLs, which is exactly how the OS/2 Workplace Shell is distributed.

THE OPENDOC PLUMBING SYSTEM

OpenDoc has the technology to revolutionize the software development

```

/*
 * This file was generated by the SOM Compiler and Emitter Framework.
 * Generated using:
 *   SOM Emitter emitxtm: 2.41
 */

#ifndef SOM_Module_divepart_Source
#define SOM_Module_divepart_Source
#endif

#define DIVEPart_Class_Source
#define VARIABLE_MACROS
#include "divepart.xih"

#define INCL_GPI
#define INCL_WIN
#define INCL_DOS

#define INCL_OS2MM
#define INCL_MMIOOS2
#define INCL_MMIO_CODEC
#define INCL_SW

#define INCL_ODARBITRATOR
#define INCL_ODCANVAS
#define INCL_ODFACET
#define INCL_ODFOCUSSET
#define INCL_ODFRAME
#define INCL_ODSHAPE
#define INCL_ODSESSION
#define INCL_ODSTORAGEUNIT
#define INCL_ODTRANSFORM
#define INCL_ODWINDOW

#include <os2.h>
#include <os2me.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <sw.h>

#define _MEERROR_H_
#include <mmioos2.h>
#include <dive.h>
#include <fourcc.h>
#include <show.h>

#ifndef _ORDCOLL_
#include "OrdColl.h"
#endif

/* New window messages to set */
/* focus to the DIVE window */
#define UM_SETFOCUSWIN (WM_USER)

/* Macro to convert a fixed */
/* value to a long */
#define FIXED2LONG(f) (((f) + 0x8000) >> 16)

/* Global variables follow */
/* PM anchor block handle */
HAB hab;

/* Image buffer numbers */
ULONG ulImage[MAX_FILE];

/* Pointer to bitmap palette */
PBYTE pPalette[MAX_FILE];

/* Flag to terminate thread */
ULONG ulToEnd = 0;

/* Location to place DIVE image */
ODPoint bottomLeft = {0, 0}, upperRight = {0, 0};

/* Bitmap filenames */
PSZ pszDefFile[] = {"TPG20000.BMP",
                    {"TPG20001.BMP",
                     {"TPG20002.BMP",
                      {"TPG20003.BMP",
                       {"TPG20004.BMP",
                        {"TPG20005.BMP",
                         {"TPG20006.BMP",

```

Listing 2. A completed program. (Continued on page 41.)



```
        {"TPG20007.BMP"},
        {"TPG20008.BMP"},
        {"TPG20009.BMP"},
        {"TPG20010.BMP"},
        {"TPG20011.BMP"},
        {"TPG20012.BMP"},
        {"TPG20013.BMP"},
        {"TPG20014.BMP"},
        {"TPG20015.BMP"} };

/* Draw count for each frame */
ULONG ulDrawCnt[] = { 1,
    1,
    1,
    1,
    1,
    1,
    1,
    1,
    1,
    1,
    1,
    1,
    1,
    1,
    1 };

/* Forward function definitions */
/* from DIVE sample */
MRESULT EXPENTRY MyWindowProc (HWND hwnd, ULONG msg,
    MPARAM mp1, MPARAM mp2);
VOID APIENTRY DirectMoveMem (ULONG parm1);
ULONG ReadFile (ULONG iFile, PSZ pszFile, PWINDATA pwinData);

/* This method is invoked when */
/* the user selects "EMBED" */
SOM_Scope void SOMLINK DIVEPartFacetAdded(DIVEPart *somSelf,
    Environment *ev,
    ODFacet* facet)
{
    HWND hwndFacet, hwndDIVE;
    DIVEPartData *somThis = DIVEPartGetData(somSelf);
    DIVEPartMethodDebug("DIVEPart", "DIVEPartFacetAdded");

    /* Add new facet to ordered */
    /* collection */
    _Facets->AddLast(facet);

    /* Retrieve facet window handle */
    if (hwndFacet = facet->GetFacetHWND(ev))
    {
        /* ALL THE FOLLOWING CODE IS */
        /* FROM THE DIVE SAMPLE -- SEE */
        /* THE SAMPLE CODE FOR COMMENTS */

        DIVE_CAPS DiveCaps = {0};
        FOURCC fccFormats[100] = {0};
        TID tidBlitThread;
        ULONG flCreate;
        ULONG i;
        PWINDATA pwinData;
        PLONG pPal;
        LONG cxWidthBorder;
        LONG cyWidthBorder;
        LONG cyTitleBar;
        LONG cyMenu;
        LONG cxWindowPos;
        LONG cyWindowPos;

        hab = 1;
        ulToEnd = 0;

        pwinData = (PWINDATA) malloc (sizeof(WINDATA));

        DiveCaps.pFormatData = fccFormats;
        DiveCaps.ulFormatLength = 120;
        DiveCaps.ulStructLen = sizeof(DIVE_CAPS);
    }
}
```

Listing 2. A completed program. (Continued on page 42.)

process. This is because OpenDoc is a platform-independent, language-independent application framework. OpenDoc's power comes from the usage of IBM's SOM technology to implement OpenDoc.

SOM provides OpenDoc with language-independence, but by itself doesn't ensure platform independence. Only the careful design of the OpenDoc framework provides this feature. OpenDoc essentially provides a complete plumbing system for an application. Figure 1 depicts a portion of the OpenDoc design. Whenever OpenDoc can provide a service that can be implemented across platforms, such as graphical clipping algorithms, it provides a working faucet that a developer can immediately unscrew. When OpenDoc runs into platform-dependent service, such as the actual drawing commands for a graphical object, it carefully caps off the pipe in such a way that a developer can easily fit a faucet of his/her choosing. OpenDoc part development tools will provide a means for designing these platform-specific faucets.

The OpenDoc plumbing system is extendible. This can be accomplished by purchasing or designing new faucets (such as three-dimensional pie charts or multimedia parts) or by purchasing or designing new extensions or modifications to the OpenDoc plumbing system, complete with faucets (word processors, spreadsheets).

Even though OpenDoc provides the complete plumbing of an application, this doesn't require that all applications have the same look and feel. This is because OpenDoc's root part (the document shell) can be replaced at startup. This gives developers the opportunity to create their own shell, customizing it as needed. Creating a shell doesn't require writing one from scratch—you can simply override the behaviors of the existing shell that you want to modify.

DIVING INTO OPENDOC

"Yeah, great—so what?" you may ask. The implications of OpenDoc cannot really be appreciated until you have developed an OpenDoc application. OpenDoc provides an entire working application wrapped in an elegant SOM framework, which is portable across multiple platforms and can be developed with multiple languages! To demonstrate the advantages acquired with OpenDoc, we will take an existing program and convert it to OpenDoc.



Without going into much detail, Direct Interface Video Extensions (DIVE) is a multimedia subsystem that provides direct access to video hardware. The DIVE APIs automatically detect and use video accelerator hardware to provide extremely fast video updates suitable for games and animation.

The DIVE sample that we will convert to OpenDoc can be found in the OS/2 Warp Developer Toolkit on the Developer Connection CD-ROM volume #7. (We will not reproduce the original source code here for brevity.) The converted program will require the OpenDoc Developer Tool-kit, OS/2 Warp Developer Toolkit, and the OS/2 Warp Fixpack #5—all on the Developer Connection CD-ROM volume #7.

The first step in developing for OpenDoc is to choose an appropriate class to subclass. This allows you to inherit behaviors (code) free of charge. Because this will be a minimal conversion to OpenDoc, I have selected the Simple class from which to start. As you develop in OpenDoc, you can reuse your existing code through subclassing.

To hook into OpenDoc, you must first use the SOM compiler to create your new class. This involves writing an IDL (Interface Definition Language) source file, which describes your new class. Remember that SOM is language neutral—the IDL is the key to this neutrality. IDL is nothing more than a class-definition language.

Our IDL, for this example, is shown in Listing 1. It defines our new class, *DIVEPart*, and the methods of the parent class which to override. The critical overrides are the *FacetAdded*, *FacetRemoved*, *GeometryChanged*, and *Draw* methods, which will process the embedding, removal, and drawing of the DIVE part.

After writing the IDL definition, invoke the SOM compiler to generate a stub file and supporting headers. The stub file contains the entry points for new methods and overrides. At first, every entry point will be either empty for new methods or contain a call to the parent method for overrides.

If we were to compile the program without modifying the C++ source file, the resulting program would function exactly as the parent Simple class. You would also find that the resulting size of the DLL would be small. This is the magic of inheritance.

Listing 2 shows the completed program with the DIVE components embedded in the appropriate over-

```
if (DiveQueryCaps(&DiveCaps, DIVE_BUFFER_SCREEN))
{
    WinMessageBox(HWND_DESKTOP, HWND_DESKTOP,
        "The sample program can not run on this system \
environment.",
        "DIVE Sample", 0, MB_OK | MB_INFORMATION);
    free(pwinData);
    return;
}

if (DiveCaps.ulDepth < 8)
{
    WinMessageBox(HWND_DESKTOP, HWND_DESKTOP,
        "The sample program can not run on this system \
environment.",
        "DIVE Sample", 0, MB_OK | MB_INFORMATION);
    free(pwinData);
    return;
}

if (DiveOpen(&(pwinData->hDive), FALSE, 0))
{
    WinMessageBox(HWND_DESKTOP, HWND_DESKTOP,
        "The sample program can not run on this system \
environment.",
        "DIVE Sample", 0, MB_OK | MB_INFORMATION);
    free(pwinData);
    return;
}

for (i = 0, pwinData->ulMaxFiles = 0; i < MAX_FILE; i++)
{
    if (ReadFile(i, pszDefFile[i], pwinData))
    {
        WinMessageBox(HWND_DESKTOP, HWND_DESKTOP,
            "Failed to open bitmaps.",
            "Error!", 0, MB_OK | MB_MOVEABLE);
    }
    else
        pwinData->ulMaxFiles++;
}

WinRegisterClass(hab, "DiveWindow", MyWindowProc, CS_MOVEONOTIFY,
    sizeof(ULONG));

pwinData->hwndFrame = WinCreateWindow(hwndFacet, "DiveWindow", NULL,
    WS_VISIBLE | WS_SYNCPAINT,
    0, 0, 0, 0,
    hwndFacet, HWND_TOP, 0, 0, 0);

pwinData->hwndClient = pwinData->hwndFrame;

WinSetWindowULong (pwinData->hwndClient, 0, (ULONG)pwinData);

pwinData->cxWidthWindow = pwinData->ulWidth;
pwinData->cyHeightWindow = pwinData->ulHeight;

pwinData->fChgSrcPalette = FALSE;

WinPostMsg(pwinData->hwndFrame, WM_VRNENABLED, 0L, 0L);

if (DosCreateThread(&(pwinData->tidBlitThread),
    (PFNTHREAD) DirectMoveMem,
    (ULONG)pwinData, 0L, 8192L))
{
    WinSetVisibleRegionNotify(pwinData->hwndClient, FALSE);

    for (i = 0; i < pwinData->ulMaxFiles; i++)
        DiveFreeImageBuffer(pwinData->hDive, ulImage[i]);

    DiveClose(pwinData->hDive);
    WinDestroyWindow(pwinData->hwndFrame);
    free(pwinData);
    return;
}
```

Listing 2. A completed program. (Continued on page 43.)


```

DosSetPriority(PRTYS_THREAD, PRTYC_IDLETIME, 0, pwinData->tidBlitThread);

/* Save the created DIVE window */
/* handle in the part's info */
facet->SetPartInfo(ev, (ODInfoType)(pwinData->hwndFrame));
/* Force a repaint of the part */
somSelf->GeometryChanged(ev, facet);
}

/* This method is called when */
/* the user either deleted the */
/* part or closes the shell */
SOM_Scope void SOMLINK DIVEPartFacetRemoved(DIVEPart *somSelf,
Environment *ev,
ODFacet* facet)
{
DIVEPartData *somThis = DIVEPartGetData(somSelf);
DIVEPartMethodDebug("DIVEPart", "DIVEPartFacetRemoved");

/* Remove the facet from the */
/* ordered collection */
if (_Facets->Contains(facet))
_Facets->Remove(facet);

HWND hwndDIVE = (HWND)facet->GetPartInfo(ev);
PWINDATA pwinData = (PWINDATA)WinQueryWindowULong(hwndDIVE, 0);

if (hwndDIVE && pwinData)
{
/* ALL THE FOLLOWING CODE IS */
/* FROM THE DIVE SAMPLE -- SEE */
/* THE SAMPLE CODE FOR COMMENTS */

int i;

ulToEnd = 1;
DosWaitThread(&(pwinData->tidBlitThread), DCWW_WAIT);

for (i = 0; i < pwinData->ulMaxFiles; i++)
DiveFreeImageBuffer ( pwinData->hDive, ulImage[i] );

DiveClose(pwinData->hDive);

if (pwinData->fccColorFormat == FOURCC_LUT8)
{
for (i = 0; i < pwinData->ulMaxFiles; i++)
DosFreeMem(pPalette[i]);
}

free(pwinData);

WinDestroyWindow(hwndDIVE);
}

DIVEPart_parent_SimplePart_FacetRemoved(somSelf, ev, facet);
}

/* Invoked when the shape of */
/* the part changes */
SOM_Scope void SOMLINK DIVEPartGeometryChanged(DIVEPart *somSelf,
Environment *ev,
ODFacet* facet)
{
ODRect rect;
DIVEPartData *somThis = DIVEPartGetData(somSelf);
DIVEPartMethodDebug("DIVEPart", "DIVEPartGeometryChanged");

/* Retrieve DIVE window handle */
HWND hwndDIVE = (HWND)facet->GetPartInfo(ev);
if (hwndDIVE)
{
MATRIXLF mtx;
ODTransform* transform = facet->GetFrameTransform(ev);

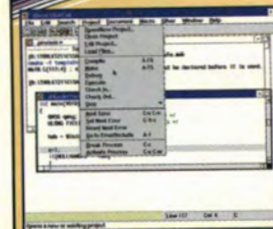
/* Ensure we can display the */
/* window */
transform->GetMATRIXLF(ev, &mtx);
if (mtx.fxM12 == 0 && mtx.fxM21 == 0 && mtx.fxM11 > 0 && mtx.fxM22 > 0)
{
/* Transform OpenDoc coords to */

```

Listing 2. A completed program. (Continued on page 44.)

VISUAL SlickEdit

THE AWARD-WINNING
PROGRAMMER'S
EDITOR



"Visual SlickEdit is the most
configurable editor I've ever seen."
"I'll pick Visual SlickEdit as the winner"
Software Development, August '95 issue

VISUAL SLICKEDIT IS THE EDITOR CUSTOM MADE
FOR YOU. ACCLAIMED AS THE MOST
CONFIGURABLE EDITOR YOU CAN BUY, VISUAL
SLICKEDIT HAS ALL THE FEATURES AND POWER
YOU CRAVE. WITH NUMEROUS AWARDS, AND
THOUSANDS OF DEVOTED USERS, VISUAL
SLICKEDIT HAS REPLACED BRIEF AS THE STANDARD
EDITOR FOR THE 90s. CALL TODAY TO GET
STARTED.

Powerful Features:

- ♦ Object oriented macro language and DLL interface
- ♦ Incremental search/search & replace
- ♦ Drag & drop editing
- ♦ Line, column, character selection
- ♦ Spell check comments & strings
- ♦ Interactive file compare
- ♦ Programmable file manager
- ♦ Macro recording

Cut your work in half using:

- ♦ Procedure tagging
- ♦ Syntax color-coding, expansion, and indenting
- ♦ SmartPaste reindents pasted or dropped source code according to nesting level
- ♦ Compiler error processing
- ♦ Entire manual on-line
- ♦ Multiple clipboards
- ♦ CUA, Brief, Emacs, and VI emulations
- ♦ Built in dialog editor
- ♦ Configurable menus and buttons

To order, Call 800-934-EDIT

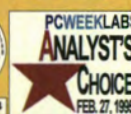
Also (919) 831-0600

FAX (919) 831-0101

72662.3312@compuserve.com



MicroEdge also makes SlickEdit available for DOS and 20
UNIX platforms. Visual SlickEdit supports C/C++, Pascal,
FORTRAN, Basic, dBASE, Module-3, Assembly,
COBOL, and Ada. The Windows NT version is available
for Intel, Mips, Alpha AXP, and Power PC machines.



MicroEdge PO Box 18038, Raleigh, NC 27619

Circle Reader Service Number 16


```

/* OS/2 PM coords */
facet->GetFrame(ev)->GetFrameShape(ev)->GetBoundingBox(ev, &rect);
ODPoint botLeft = {rect.left, rect.bottom},
topRight = {rect.right, rect.top};

bottomLeft.x = rect.left;
bottomLeft.y = rect.bottom;
upperRight.x = rect.right;
upperRight.y = rect.top;

botLeft = transform->TransformPoint(ev, &botLeft);
topRight = transform->TransformPoint(ev, &topRight);
/* Position the DIVE window */
WinSetWindowPos(hwndDIVE, 0,
FIXED2LONG(botLeft.x),
FIXED2LONG(botLeft.y),
FIXED2LONG(topRight.x - botLeft.x),
FIXED2LONG(topRight.y - botLeft.y),
SWP_MOVE | SWP_SIZE);
/* Calculate the clip region */
ODShape* clipShape = facet->GetAggregateClipShape(ev)->Copy(ev);
clipShape->Transform(ev, facet->GetWindowFrameTransform(ev));
ODTransform* offset = new ODTransform;
botLeft.x = -botLeft.x;
botLeft.y = -botLeft.y;

offset->SetOffset(ev, &botLeft);
clipShape->Transform(ev, offset);
/* Clip the DIVE window */
WinSetClipRegion(hwndDIVE, clipShape->GetRegion(ev));
delete clipShape;
WinShowWindow(hwndDIVE, TRUE);
WinPostMsg (hwndDIVE, WM_VRENABLED, 0L, 0L );
}
else
{
WinShowWindow(hwndDIVE, FALSE);
}
}
}

/* Invoked for user-initiated */
/* events */
SOM_Scope ODBoollean SOMLINK DIVEPartHandleEvent(DIVEPart *somSelf,
Environment *ev,
ODEventData* event,
ODFrame* frame,
ODFacet* facet)
{
DIVEPartData *somThis = DIVEPartGetData(somSelf);
DIVEPartMethodDebug("DIVEPart", "DIVEPartHandleEvent");

/* Set the focus to the DIVE */
/* window if requested */
if (facet && event->hwnd == facet->GetFacetHWND(ev))
{
if (event->msg == UM_SETFOCUSWIN)
{
WinSetFocus(HWND_DESKTOP, (HWND)event->mp1);
}
}

return (DIVEPart_parent_SimplePart_HandleEvent(somSelf, ev,
event, frame,
facet));
}

/* Invoked when part needs to */
/* be initialized */
SOM_Scope void SOMLINK DIVEPartInitPart(DIVEPart *somSelf,
Environment *ev,
ODStorageUnit* storageUnit)
{
DIVEPartData *somThis = DIVEPartGetData(somSelf);
DIVEPartMethodDebug("DIVEPart", "DIVEPartInitPart");

/* Only initialize once */
if (somSelf->IsInitialized(ev))
return;

```

Listing 2. A completed program. (Continued on page 45.)



```
somSelf->InitPersistentObject(ev, storageUnit);
/* Create the ordered collection*/
/* to hold facets */
_Facets = new ODxOrderedCollection;
}

/* Called when the document is */
/* loaded by OpenDoc */
SOM_Scope void SOMLINK DIVEPartInitPartFromStorage(DIVEPart *somSelf,
                                                    Environment *ev,
                                                    ODStorageUnit* storageUnit)
{
    DIVEPartData *somThis = DIVEPartGetData(somSelf);
    DIVEPartMethodDebug("DIVEPart", "DIVEPartInitPartFromStorage");

    /* Only initialize once */
    if (somSelf->IsInitialized(ev))
        return;

    somSelf->InitPersistentObjectFromStorage(ev, storageUnit);
    /* Create the ordered collection*/
    /* to hold facets */
    _Facets = new ODxOrderedCollection;

    DIVEPart_parent_SimplePart_InitPartFromStorage(somSelf, ev, storageUnit);
}

/* Called to allow the part to */
/* do platform-specific drawing */
SOM_Scope void SOMLINK DIVEPartDraw(DIVEPart *somSelf,
                                     Environment *ev,
                                     ODFacet* facet,
                                     ODShape* invalidShape)
{
    DIVEPartData *somThis = DIVEPartGetData(somSelf);
    DIVEPartMethodDebug("DIVEPart", "DIVEPartDraw");

    /* Get DIVE window handle */
    HWND hwndDIVE = (HWND)facet->GetPartInfo(ev);
    if (hwndDIVE && WinIsWindowShowing(hwndDIVE))
    {
        /* Drawing is done by the DIVE */
        /* thread */
        WinPostMsg (hwndDIVE, WM_VRNENABLED, 0L, 0L );
    }

    /* Updates facets as needed */
    SOM_Scope void SOMLINK DIVEPartUpdateFacets(DIVEPart *somSelf,
                                                Environment *ev,
                                                ODFacet* facet)
    {
        DIVEPartData *somThis = DIVEPartGetData(somSelf);
        DIVEPartMethodDebug("DIVEPart", "DIVEPartUpdateFacets");

        HWND hwndDIVE = facet->GetPartInfo(ev);

        ODxOrderedCollectionIterator facets(_Facets);
        ODFacet* f = (ODFacet*) facets.First();
        while (facets.IsNotComplete() == kODFalse)
        {
            if (f != facet)
            {
                HWND hwnd = (HWND)f->GetPartInfo(ev);
                if (!hwnd || !WinIsWindowShowing(hwnd))
                {
                    f->Invalidate(ev, kODNULL);
                }
            }
            f = (ODFacet*) facets.Next();
        }
    }

    /* Invoked when the user activates*/
    /* the part */
    SOM_Scope void SOMLINK DIVEPartActivateFrame(DIVEPart *somSelf,
                                                Environment *ev,
                                                ODFrame* frame)
    {
        ODSession *fSession;
    }
}
```

rides. The `FacetAdded` method is called by `OpenDoc` whenever the user embeds a part, so I have placed the creation of the DIVE PM window here and the startup of the animation. `FacetRemoved` is invoked when the part is deleted, so I must stop the animation and remove the DIVE window.

The `GeometryChanged` method controls part movement and resizing by the user. Note that here we must convert the platform-independent coordinates provided by `OpenDoc` into platform-specific coordinates for use by PM. `OpenDoc` provides the necessary transformation matrices to do the dirty work for us.

The dirty work of actually doing the DIVE screen updates is processed in a second thread, which was spawned when the part was embedded. Therefore, the overridden `Draw` method needs to only tell the secondary thread to refresh the screen.

That's it! With the exception of converting `OpenDoc` coordinates to PM coordinates, not much has been added to accomplish the conversion. In fact, the sample lost a few pounds of code in the process. This is because `OpenDoc` already handles a lot of low-level dirty work such as starting a PM message processing loop. What remains is the heart of the application, the actual DIVE commands controlling the screen updates. This is the future of application development—focusing on your particular area of expertise, not on the periphery code, which is tedious but required.

If DIVE were available on other platforms, this sample could also function on those with minimal modifications. Assuming the DIVE API was portable, only the PM APIs and the thread creation APIs would need to be recoded. That's quite a bang for the buck!

This fully functioning example, complete with a three-dimensional jet animation, can be downloaded from the OS2DF2 CompuServe forum. Look for the file DIVE.ZIP in LIB 13. This sample will also be available on a future Developer Connection CD-ROM.

CONCLUSION

A typical method of starting development on a program is to use a working template as the starting point. Most development toolkits provide the source code for dozens of working programs, which can be used as working templates to start development. This provides the developer with working functionality, which can be


```

ODTypeToken      fSelectionFocus;
DIVEPartData *somThis = DIVEPartGetData(somSelf);
DIVEPartMethodDebug("DIVEPart", "DIVEPartActivateFrame");

DIVEPart_parent_SimplePart_ActivateFrame(somSelf, ev, frame);

fSession = somSelf->GetStorageUnit(ev)->GetSession(ev);
fSelectionFocus = fSession->Tokenize(ev, kODSelectionFocus);

if (fSession->GetArbitrator(ev)->GetFocusOwner(ev, fSelectionFocus) != frame)
{
    ODFrameFacetIterator* facets = frame->CreateFacetIterator(ev);
    ODFacet* facet = facets->First(ev);
    delete facets;

    HWND hwndDIVE = (HWND)facet->GetPartInfo(ev);
    HWND hwndFocus = WinQueryFocus(HWND_DESKTOP);

    if (hwndFocus == hwndDIVE)
    {
        HWND hwndShell = WinWindowFromID(facet->GetWindow(ev)->
                                         GetPlatformWindow(ev), FID_CLIENT);
        WinPostMsg(facet->GetFacetHWND(ev), UM_SETFOCUSWIN,
                  (LPARAM)hwndShell, 0);
    }
}

/* Invoked when the user activates */
/* the part */
SOM_Scope void SOMLINK DIVEPartActivatingWindow(DIVEPart *somSelf,
                                                  Environment *ev,
                                                  ODFrame* frame)
{
    ODSession      *fSession;
    ODTypeToken      fSelectionFocus;
    DIVEPartData *somThis = DIVEPartGetData(somSelf);
    DIVEPartMethodDebug("DIVEPart", "DIVEPartActivatingWindow");

    DIVEPart_parent_SimplePart_ActivatingWindow(somSelf, ev, frame);

    fSession = somSelf->GetStorageUnit(ev)->GetSession(ev);
    fSelectionFocus = fSession->Tokenize(ev, kODSelectionFocus);

    if (fSession->GetArbitrator(ev)->GetFocusOwner(ev, fSelectionFocus) == frame)
    {
        ODFrameFacetIterator* facets = frame->CreateFacetIterator(ev);
        ODFacet* facet = facets->First(ev);
        delete facets;

        HWND hwndDIVE = (HWND)facet->GetPartInfo(ev);
        HWND hwndFocus = WinQueryFocus(HWND_DESKTOP);

        if (hwndDIVE && WinIsWindowShowing(hwndDIVE) && hwndFocus != hwndDIVE)
        {
            WinPostMsg(facet->GetFacetHWND(ev), UM_SETFOCUSWIN,
                      (LPARAM)hwndDIVE, 0);
        }
    }
}

/* Invoked when the user activates */
/* the part */
SOM_Scope void SOMLINK DIVEPartFocusAcquired(DIVEPart *somSelf,
                                              Environment *ev,
                                              ODTypeToken focus,
                                              ODFrame* ownerFrame)
{
    ODSession      *fSession;
    ODTypeToken      fSelectionFocus;
    DIVEPartData *somThis = DIVEPartGetData(somSelf);
    DIVEPartMethodDebug("DIVEPart", "DIVEPartFocusAcquired");

    DIVEPart_parent_SimplePart_FocusAcquired(somSelf, ev, focus, ownerFrame);

    fSession = somSelf->GetStorageUnit(ev)->GetSession(ev);
    fSelectionFocus = fSession->Tokenize(ev, kODSelectionFocus);

    if (focus == fSelectionFocus)

```

Listing 2. A completed program. (Continued on page 47.)



OS/2® and AIX Tools at Your Command

The best way to
find out what's
available for
IBM system
software

platforms.

Stay
on top of
it by browsing
this software and
services directory of
products for OS/2,
AIX and other
IBM supported
operating systems.

Sponsored by IBM®, OS/2 Developer,
OS/2 Magazine, and Software Development.

OS/2® is a registered trademark of
International Business Machines
Corporation and is used by Miller
Freeman, Inc. under license.

**Find It
on the
Web!**

<http://www.mfi.com/softwareguide>

```
{
    HWND hwndFocus = WinQueryFocus(HWND_DESKTOP);

    ODFacetIterator* facets = ownerFrame->CreateFacetIterator(ev);
    ODFacet* facet = facets->First(ev);
    delete facets;

    HWND hwndDIVE = (HWND)facet->GetPartInfo(ev);
    /* This will pause the animation*/
    WinPostMsg (hwndDIVE, WM_VRNDISABLED, 0L, 0L );

    if (hwndDIVE && WinIsWindowShowing(hwndDIVE) && hwndFocus != hwndDIVE)
    {
        WinPostMsg(facet->GetFacetHWND(ev), UM_SETFOCUSWIN,
            (MPARAM)hwndDIVE, 0);
    }
}

/* Invoked when the user */
/* deactivates the part */
/*
SOM_Scope void SOMLINK DIVEPartFocusLost(DIVEPart *somSelf,
    Environment *ev,
    ODTypeToken focus,
    ODFrame* ownerFrame)
{
    ODSession *fSession;
    ODTypeToken fSelectionFocus;
    DIVEPartData *somThis = DIVEPartGetData(somSelf);
    DIVEPartMethodDebug("DIVEPart", "DIVEPartFocusLost");

    DIVEPart_parent_SimplePart_FocusLost(somSelf, ev, focus, ownerFrame);

    fSession = somSelf->GetStorageUnit(ev)->GetSession(ev);
    fSelectionFocus = fSession->Tokenize(ev, kODSelectionFocus);

    if (focus == fSelectionFocus)
    {
        ODFacetIterator* facets = ownerFrame->CreateFacetIterator(ev);
        ODFacet* facet = facets->First(ev);
        delete facets;

        HWND hwndDIVE = (HWND)facet->GetPartInfo(ev);
        /* Resume the animation */
        WinPostMsg (hwndDIVE, WM_VRNNENABLED, 0L, 0L );

        HWND hwndFocus = WinQueryFocus(HWND_DESKTOP);
        if (hwndFocus == hwndDIVE)
        {
            HWND hwndShell = WinWindowFromID(facet->GetWindow(ev)->
                GetPlatformWindow(ev), FID_CLIENT);
            WinPostMsg(facet->GetFacetHWND(ev), UM_SETFOCUSWIN,
                (MPARAM)hwndShell, 0);
            WinSetFocus(HWND_DESKTOP, hwndShell);
        }
    }
}

/* Invoked by SOM when this */
/* class is initialized */
/*
SOM_Scope void SOMLINK DIVEPartsomInit(DIVEPart *somSelf)
{
    DIVEPartData *somThis = DIVEPartGetData(somSelf);
    DIVEPartMethodDebug("DIVEPart", "DIVEPartsomInit");

    DIVEPart_parent_SimplePart_somInit(somSelf);
    /* Initialize ordered collection*/
    /* pointer */
    _Facets = kODNULL;
}

/* Invoked by SOM when this */
/* class is un-initialized */
/*
SOM_Scope void SOMLINK DIVEPartsomUninit(DIVEPart *somSelf)
{
    DIVEPartData *somThis = DIVEPartGetData(somSelf);
    DIVEPartMethodDebug("DIVEPart", "DIVEPartsomUninit");

    DIVEPart_parent_SimplePart_somUninit(somSelf);
}
```

Listing 2. A completed program. (Continued on page 48.)



augmented until it becomes a finished application. Usually, however, the samples are severely limited in scope and tend to provide only a single working piece of functionality, such as a bit map viewer or a menu bar. To make a fully functioning application usually requires the splicing together of many provided samples.

OpenDoc can change the way applications are designed, programmed, maintained, and used for the better. You can now disassociate OpenDoc from compound documents and think of it more as a starting point for application development.

OpenDoc has the technology to not only become the application that OS/2 users have been waiting for, but also the application framework, which OS/2 developers have been craving. Hope I was able to keep you afloat from this DIVE into OpenDoc!

Robert Yeager has been an SOM developer since its introduction in OS/2 2.0. He was the co-founder of Synaptec, the leading provider of SOM development tools. He is currently finishing work on *The Object Factory for Warp V. 3.0*, which features a visual SOM Interface Repository browser and IDL importation capability. Bob has a B.S. from the University of Evansville and an M.S. from Central Michigan University.

```

delete _Facets;
}

/* Delete the ordered collection*/

/* THE FOLLOWING DIVE FUNCTIONS */
/* HAVE NOT BEEN MODIFIED FROM */
/* THE DIVE SAMPLE AND ARE NOT */
/* DUPLICATED HERE */

VOID APIENTRY DirectMoveMem (ULONG parm1)
{
}

ULONG ReadFile (ULONG iFile, PSZ pszFile, PWINDATA pwinData)
{
}

MRESULT EXPENTRY MyWindowProc (HWND hwnd, ULONG msg, MPARAM mp1, MPARAM mp2)
{
}

```

Listing 2. A completed program. (Continued on page 48.)

REFERENCES

Orfali, Robert, and Dan Harkey. *The Client/Server Survival Guide with OS/2*, Van Nostrand Reinhold, 1994, ISBN 0-442-01798-7.

Lau, Christina. *Object-Oriented Programming Using SOM and DSOM*, Van Nostrand Reinhold, 1994, ISBN 0-442-01948-3.

Tate, Bruce, Danforth, Scott, and Paul Koenen. *Objects for OS/2*, Van Nostrand Reinhold, 1994, ISBN 0-442-01738-3.

The Developer Connection for OS/2. The OS/2 Developer Connection CD-ROM can be ordered by phone at (800) 633-8266 or fax (303) 330-7655.

Did You Miss These?

Back Issues

To order your back issues of OS/2 Developer, call now:

800-WANT-OS2 (800-926-8672)

or write to:

OS/2 Developer, P. O. Box 1079
Skokie, IL 60076-8079

Issues are \$9.95 plus \$2.95 shipping and handling.
All orders must be prepaid.

Hurry!
Supplies Limited!

A Special Report from the publishers of OS/2 Developer!

Quantities are limited.
Order now!
For fastest service,
fax your order to us
at 415-905-4967.



Smalltalk

☒ **YES!** Please send me _____ copies of Smalltalk!

Each copy is only \$9.95 in the US or \$12.95 in Canada.

Please add \$2.50 for shipping and handling.

- ☐ My check is enclosed
- ☐ Charge my credit card: ☐ Visa ☐ MasterCard ☐ American Express

Name _____

Signature _____

Date _____

Send them to:

Name _____

Address _____

City _____

State _____

Postal Code _____

Mail: Smalltalk
P. O. Box 7046
San Francisco
CA 94120

Phone: 1-800-444-4881

Fax: (415) 905-4967



Normal Form and Other Mantras of Effective Database Design

A database, according to the model I have grown up with, in its simplest form, is a two-dimensional array usually characterized as a table of rows and columns commonly represented by a spreadsheet. The rows represent so-called entities, and the columns are attributes with which the entities are endowed. A simple example is a mailing database, which contains names and addresses for generating labels and possibly includes other information about the named individuals that can be used for selection purposes. In spreadsheet form, each row corresponds to one label and the columns contain the separate bits of required data: names, street addresses, cities, states, zip codes, and perhaps phone numbers, business addresses, and other facts about the people represented. For example, a university's alumni mailing list might include the year of graduation, degrees, honors, and whether the current dues to the Alumni Association are in arrears.

In the current frenzy for multimedia,

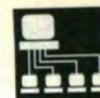
software object orientation, and World Wide Web hypertext links, this simple model might need to be enhanced in various ways. I'm inclined to argue, however, that its underpinnings remain substantially unchanged. One might add to the attributes an image depicting the alumnus/a at the graduation ceremony, a sound byte from the phone call that explains how the check for the current dues payment is in the mail, or even drop in a pointer to a folder that includes yet other objects associated with this person. Nevertheless, we can still work from the spreadsheet model in which these additional attributes are handled by new columns with pointers linking the entity to these various objects.

This brings us to our first design topic: refining table structure. Upon closer examination of the mailing database, you may notice that each entity has been endowed with a certain amount of redundant information that might be worth eliminating. A simple example is the state of residence and the zip code. Perhaps, out of a desire to sustain the

moral fabric of America, postal zip codes are not permitted to cross state lines. Accordingly, if I know people's zip codes, then I know the state in which they live. I could, therefore, eliminate the state field from the records in my mailing list by adding to the database a separate table in which zip codes and states are matched. To print the address, the label procedure first checks the zip code, then goes to

Last Name	First Name	Address	City	St	ZipCode	Are	Phone
Adams	Joan	Apartment 12	St. Louis	MO	63119-1234		-
Addison	Bill	1775 Horder Roa	Chesterfield	MO	63005-1234	314	555-18
Adkisson	Kelly	346 Connecticut	St. Louis	MO	63116-0000	314	555-10
Adrian	Thomas	332 Tree Lane	Florissant	MO	63034-1234	314	555-59
Adrian	Marc	13076 Bob Dr.	St. Louis	MO	63135-0000	314	555-30
Ahrens	Jodi	404 Shirley Pla	St. Louis	MO	63123-0000	314	555-87
Ahrens	Sally	1074 Whole Driv	St. Louis	MO	63128-1234	314	555-44
Albaugh	Alice	2456 Florissant	Normandy	MO	63121-1234	314	555-38
Albers	Albert	13 Sheep Meadow	St. Louis	MO	63127-1234	314	555-69
Billy	Billy	5200 Queenswood	St. Louis	MO	63123-1234	314	555-81
Albrecht	Jason	2385 Aprilrook	Kirkwood	MO	63122-1234		-
Aleman	Andy	4009 Elm Avenue	East St. Lou	IL	62201-1234	618	555-22
Allen	Walter	717 Eldon Avenue	Maryland Hei	MO	63043-1234	314	555-20
Anastas	Harry	139 Forest	St. Louis	MO	63131-1234	314	555-87
Anthony	Peter	169 Heather Moo	Florissant	MO	63034-0000	314	555-17
Appelbaum	Samual	730 Mary Court	St. Louis	MO	63125-1234	314	555-42
Appleby	Jackson	3120 Toll Road	St. Louis	MO	63129-1234		-
April	Daniel	17127 Sulley Vi	Chesterfield	MO	63005-1234	314	532-02
Arends	Terry	2300 Coventry C	Chesterfield	MO	63017-1234	314	230-37
Armbruster	Oscar	20 Suns	Creve Coeur	MO	63141-1234	314	432-88

Figure 1. Part of the original, unrefined alumni mailing list.



the new zip/state table to retrieve the two-character state code. It probably takes longer, but it's much more elegant and, more important, it dramatically enhances data management. Okay, so there's not much management enhancement here, but consider the following similar scenario.

Suppose, for the sake of this illustration, the U.S. Postal Service and Ma Bell decide to give zip codes and area codes coinciding boundaries. In your mailing database, you eliminate the redundancy by creating a table that links area and zip codes. If you know the zip code, you have the area code because no zip code region crosses area code boundaries. Now, as it happens, the phone company is running out of numbers in the present area code blocks, and they plan to realign them. Suddenly, 5,000 records in your mailing database must have the area code field changed to reflect the new assignments. So, which method will be the fastest and most accurate: changing 5,000 records individually and worrying about whether you forgot or made any mistakes, or going to the area/zip code table and changing just the zip code records that have received new area code assignments? Therein lies the gain, which is to be had by building so-called relational or hierarchical databases with linked tables joining shared information.

NORMAL FORM

The type of construct we have just reviewed is formalized in a set of standards called Normal Form. Normalizing a database ensures that the attributes associated with any particular table belong only to the entities singled out in the records of that table. In less jargon-laden terms, normalizing a database makes sure that each column in the table's spreadsheet says something unique and specific about the occupant of each row. In addition, Normal Form sets standards for the relationships that may be defined between one table and another. These standards go well beyond the simple removal of redundancy

within records. The rules for their implementation are sometimes written in a language near and dear to the theoretical mathematician, so in the interest of clarity, I will endeavor to paraphrase and offer some illustrations.

The mailing database currently has two tables: the alumni table, which uses something unique to the individual like a Social Security number as a key to identify each record, and the zip/area code table, with zip code as its key. The two tables are joined on the zip code field, which they share. For the sake of the following examples, assume that the alumni table also has fields for degree earned and in what year, the major area of study, and up to three past and present employers along with hire dates and job titles.

The head of the Alumni Association has decided that the mailing list should be expanded into a more comprehensive database. You review the records, and your attention is drawn to the repeating groups of employers, hire dates, and job classifications. This violates the stated goal of making each assigned attribute unique. We currently have three different sets of attributes providing inherently similar information about each entry in the mailing list. It makes more sense to separate these into a table of their own, with a separate record for each employer. The key will be a combination of two fields, the alums' Social Security number, identifying the corresponding record in the alumni table, and the name of the employer. The additional attributes are the hire date, the job title, the degree earned at the school, and the major

Last Name:	[Redacted]	
First Name:	[Redacted]	
Address:	[Redacted]	
City:	[Redacted]	
State:	[Redacted]	
ZipCode:	[Redacted]	
Area Code:	[Redacted]	
Phone Number:	[Redacted]	
School:	[Redacted]	
Degree:	[Redacted]	
Year Graduated:	[Redacted]	
Major Subject:	[Redacted]	
Alumni Dues Paid to:	[Redacted]	
Current Employer:	[Redacted]	
Job Title:	[Redacted]	
Hire Date:	[Redacted]	
Previous Employer:	[Redacted]	
Previous Job Title:	[Redacted]	
Previous Hire Date:	[Redacted]	
Next Previous Employer:	[Redacted]	
Next Previous Job Title:	[Redacted]	
Next Previous Hire Date:	[Redacted]	

Figure 2. The fields as they appear in the non-normalized table.

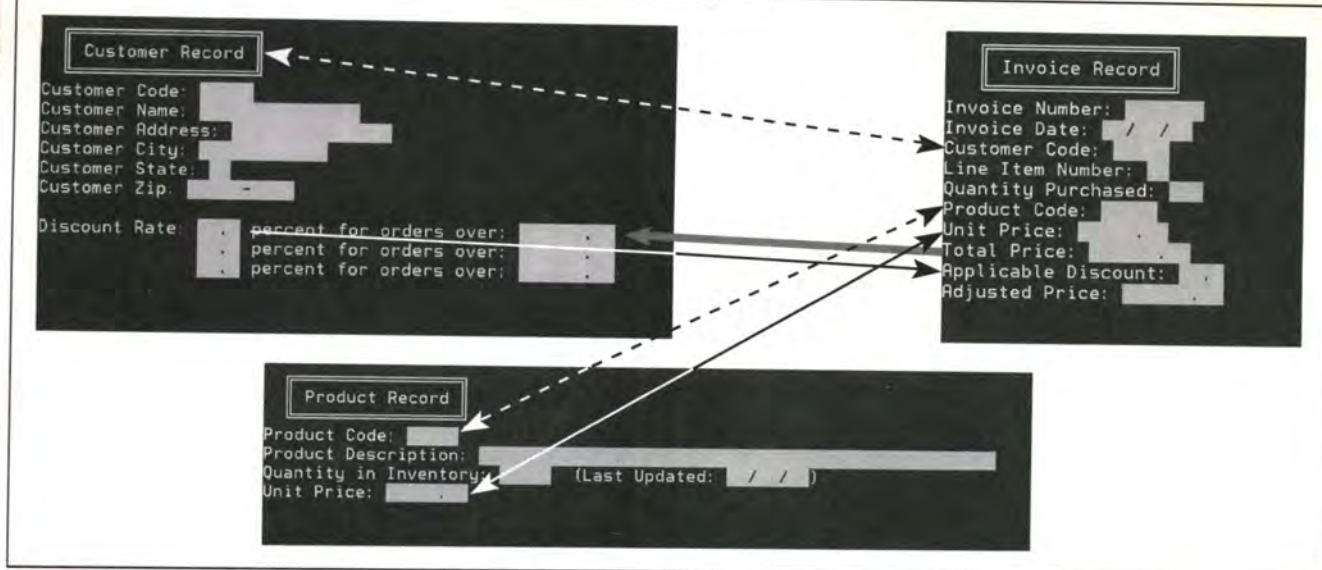


Figure 3. The three linked record types in the billing system.

subject area of study. The latter fields have been included to indicate the suitability of the educational experience to the job currently held (or not, as the case may be), such as the CEO has an MBA.

A second set of potentially repeating groups is the degree received and the year of graduation. While most received only a single degree,

some may have stayed on or returned for one or more advanced degrees. A separate table, keyed jointly on alumni code and degree with graduation date as the remaining attribute, would take care of this repetition.

The elimination of these explicit or potential repeating groups places the alumni table into First Normal Form. It has the further advantage of

more easily adapting to situations where there are more than three employers to track.

In taking a second look at the alumni-employer cross-reference table, there appears to be something not quite right about having duplicated the degrees and subject majors in each of these records. While including this information may have some

You wouldn't use a light-weight development tool to build a business-critical system. So why would you use a light-weight testing tool to test it?

The Automated Test Facility was built to test the kinds of client/server systems that go directly to a company's bottom line: customer service, claims adjustment, loan processing, inventory management. Systems where reliability and quality are essential.

ATF's unique ability to do multi-user workflow testing has made ATF the testing tool of choice for corporations building business-critical systems under OS/2, Windows, and NT.

Want to hear more? Call 617-576-2257 (or fax 617-864-7747).



**The Softbridge
Automated Test Facility**

Softbridge, Inc. ▲ 125 CambridgePark Drive ▲ Cambridge MA 02140

Circle Reader Service Number 17

© 1995 Cirrus Technology

UniteObjects

for OS/2 Warp

Unite® Scanner Object

Add color, grayscale and bitonal scanning to your applications. Elegant interface supports personal to 110 PPM scanners.

► **SOM Based**

► **32-bit APIs** accessible from C, C++, VX-REXX™, VisualAge™...

► **Beta OpenDoc™** Unite parts included

Unite® Image Viewer Object

Add image viewing, manipulating and faxing to your applications. Compatibility with many file formats.

Unite® Storer Object

Storage and retrieval for optical and CD-recordable jukeboxes. Multi-threaded design for superior performance. OS/2 Warp device drivers included.

1-800-272-1135

Cirrus Technology, 5301 Buckeystown Pike, Frederick, Maryland 21704

Circle Reader Service Number 18

FORM RELATIONSHIP

BETWEEN

Form 1: Invoice

If Form 1 is Subform, how should Key field be updated when Main form Key is modified: Null (Blank)

& Form 2: Product

If Form 2 is Subform, how should Key field be updated when Main form Key is modified: Restrict (Don't Update)

BASED ON THE FOLLOWING FIELDS BEING EQUAL: (Define at least one set)

Field	Product Code	=	Product Code
And Field		=	
And Field		=	

OPTIONAL RELATIONSHIP NAMES:
(Form names are used as a default.)

for Form 1: Form 2: Lookup Product Code

Figure 4. DataEase maintains referential integrity when defining the relationship between product and invoice.

merit for the university's image, it is logically unsound from a design perspective. The hire date and job title are attributes of both the individual and the indicated company taken together, and hence inform about the entire record key, which combines both the individual and employer fields. The degree and area major say something only about the individual, not about the company. Remembering that a goal of Normal Form is to achieve attributes both unique and

specific, these two attributes should properly (or normally) remain back in the alumni table, where they are specific to the individual members, instead of being inserted into the alumni-employer cross reference.

Returning the degree and major to the alumni table places the cross-reference table in Second Normal Form. All attributes are associated with the entire compound key, not merely with some parts of that key.

The alumni table is in First

Normal Form because we eliminated repeating groups and in Second Normal Form by default (because it has no compound key). The elimination of the zip/area code redundancy has also placed it in Third Normal Form. Given the hypothetical scenario sketched out earlier, the area code has effectively become an attribute of the zip code and is no longer specific to the alumni list member. The goal of Third Normal Form is to remove overlapping and, hence, redundant attributes within a table, consigning them to tables of their own where the data can be more easily managed. In essence, without the introduction of the area code/zip code table, we have two attributes doing the work in a table where one will suffice.

There are also Fourth and Fifth Normal Form standards that relate to the isolation of independent or semantically related multiple relationships.

REFERENTIAL INTEGRITY

Normal Form is designed to protect information by splitting it up in various ways. For example, in the alumni database, it might be advisable to break out employers into their own table instead of confining them to the

IBM LAN Server? You Must Have "LAN Intensive Care Utilities"

Without it, what if...

- Alias logon assignments fail?
- Domain/accounts corrupted?
- Permission disappear?

How do you...

- Break-up/consolidate domains?
- Move accounts/resources?
- Administer hundreds of accounts?

You're dead without our tools!

Lieberman and Associates
221 N. Robertson Blvd. Suite C
Beverly Hills, CA 90211-1703
Phone: 310-550-8575
Fax: 310-550-1152 BBS: 310-550-5980
CS: 76426,363 IBMMAIL: USMVHLVH
Call: 800-829-6263

Try it out yourself today!
Download it from: Our BBS,
CompuServe: GO OS2DF2 Lib3,
IBMLINK, Hobbes



See us at OS/2
World in
Boston, MA
Booth 706

CALL OR
FAX FOR
IMMEDIATE
INFO!

Imaging Object Pack™ for VX•REXX and C/C++



The Imaging Object Pack enables you to incorporate powerful image processing into your VX•REXX and C/C++ projects. Using the **Standard Edition** you can read, write, and display images of various formats (BMP, GIF, PCX, TGA, TIF and more).

The **Professional Edition** provides advanced functionality which includes JPEG support, image rotation, reflection, transposition, bits per pixel, palette, and color space conversions, halftone, error diffusion, printing, animation and more.

Order now and receive the Imaging Object Pack at these special introductory prices (limited time pricing includes royalty-free distribution rights):

Standard Edition \$49.95
Professional Edition \$149.95

CALL: (800) 830-8703
FAX: (941) 377-7392

 **DB Technologies, Inc.**
2420 Briar Oak Circle
Sarasota, Florida 34232-6100

Internet: dbtech@ibm.net
CompuServe: 72123,3661

VX•REXX is a trademark of the Watcom International Corporation.



alumni-employer cross reference. If the last alum who worked for a given company dies and is removed from the records, all record of that company goes too. But, splitting up information is not without its adverse consequences, which brings us to our second topic—referential integrity. As soon as you decide to distribute information among multiple tables, you have to deal with what happens to the data in one table when you make changes in another.

To change illustrations, a billing system contains a table of customer information, including discount rates based on volume, a product table, and an invoice table containing records of current and past transactions. The price paid on the invoice is calculated from the list prices, taken from the product table, and the discount rate from the customer table.

What happens in the invoice table when there is a change in the product table's list price or in a customer's discount rate? Common sense dictates that the change should only be applicable to invoice records created after the date of the revision. This can be handled by incorporating the revision date in the "parent" customer or product record and using a compound key, which contains the new rate or price, plus an effective date to define the affected invoice records. Easy enough. On the other hand, if business is good and it becomes necessary to expand the number of characters in the customer account key, such a change should be cascaded through all the associated invoices and any other tables containing customer-related information. Last, let's suppose a customer leaves the fold and the corresponding customer record is deleted. Should the corresponding invoice records be deleted at the same time, archived, or left in their original locations in the invoice table? The challenge for the developer lies in identifying all of these potential scenarios during discussions with the client user and then including the required fields and associated routines to cover them. However, equally important is that the database development tool chosen for the project is, shall we say, robust enough to accommodate these demands.

DATAEASE

There aren't a great many development tools written specifically for OS/2 or tools that have been ported over successfully from either the DOS or Windows arena, but one interest-

ing exception is DataEase, a product from DataEase International Inc. DataEase has had OS/2 versions dating back to the days of the 1.x versions and, only recently (with release 5.0), has marketed a complete Windows offering. DataEase supports multi-user applications through network file- and record-locking features. Through a separate product, SQL Connect, DataEase applications can be defined in a client or server environment, using the same client side features that are available for desktop and LAN-based projects. Over the years, DataEase has remained a solid development tool, which has not only won a following in the user market but has also formed the basis of several commercial applications.

I cite DataEase here because, in addition to offering a native OS/2 version, it is a tool in which I have done the majority of my database development projects. It was not a product of choice, but one already in place when I joined the law firm where I work. Nevertheless, having done maintenance work in other systems, DataEase is the product I always return to for new applications. As its name is intended to suggest, it is extremely easy to build quick and clean prototype systems. Where I work, litigators are busy with the process of discovery, and rarely discover the need for data-processing support until the eleventh hour. This sometimes means that I hear about the needed application on a Friday, and it must be finished and into production on the following Monday.

DataEase incorporates a number of built-in features that assist in achieving the standards of normal form and referential integrity. It thus serves as an illustration of how these requirements can be enforced in applications that are written using third-party development software.

LINKING TABLES

With respect to referential integrity, DataEase links tables with a simple method that uses up to three related fields. When defining a relationship between two tables, the designer has the option of specifying how changes in the key fields defining the linkage should be handled.

The choices are Cascade, Null, and Restrict. Cascade means that child records will be changed to reflect changes in the parent and are deleted when the parent record is deleted. In other words, modify or

lose the customer and the invoices are modified or deleted as well.

Null means that the corresponding key value in the child records will be set to BLANK when the value in the parent is changed. The child records are still deleted along with the parent. The advantage of Null is to purposely orphan the child records so that a follow-up can isolate them and process them appropriately.

Restrict means that changes in the parent, including deletion, leave the child records unaffected. In SQL Connect, two additional integrity rules are included to enforce the Cascade and Null options on the server engine.

COMPOUND INDEXING

In earlier versions, DataEase wasn't as congenial to the rules of Normal Form as it has become in the current release. This may seem like an odd comment, since Normal Form appears to be inherently a design issue without regard to the chosen development tool. True enough. But realizing the chosen design, without making compromises or sacrificing performance, depends on the underlying software. By way of example, enforcing the first and second rules of Normal Form can be enhanced by the ability to create and maintain compound indexes over multiple fields in a record table.

In earlier versions of DataEase, indexing was restricted to individual fields. To build a compound index required the creation of an indexed "dummy" field made up by joining the content of the fields logically connected as the record key. The "join-text" function that accomplished this, as its name suggests, converted field values of distinct types into a common text format. This loss of the original field characteristics frequently proved impractical for purposes of linking records into their required relationships.

The current 5.0 release now supports the creation of true compound indexing, incorporating fields with distinct type characteristics. There are, however, two limitations. If you are building tables that will be accessed from the DataEase for Windows software, as well as from OS/2 or DOS, tables with compound indexes will be viewed as read-only from that perspective. Users will be able to view but cannot create or modify the associated records. Additionally, compound indexes cannot be defined on SQL or other for-

eign database tables. At least it's a step in the right direction.

In choosing a development tool for application building in the OS/2 operating environment, your decision ought to be based at least in part on how well and how easily the product enforces the design standards you should be following. If you are thinking about checking out DataEase, one downside for OS/2 developers is that DataEase for OS/2 is still text-based like its DOS cousin. On the other hand, with a few exceptions like the one noted above, the file formats for the 5.0 DOS, Windows, and OS/2 releases are fully equivalent. This can offer an advantage in a setting where the users may have a variety of front-end operating environments on their desktops.

Charles R. Meyer, who holds a Ph.D. in logic and philosophy of science, is senior PC applications specialist with the law firm of Bryan Cave LLP. Charles has been part of the firm's MIS department, based in St. Louis, MO, for eight years, and is responsible for programming and database development in support of both administrative and case management projects. Prior to that he worked as a programmer and independent consultant and has also taught database design and programming.

REFERENCES

Date, C. J. *An Introduction to Database Systems, Volume 1, fifth edition*. Addison-Wesley Publishing Co. Inc., 1990.

Date provides a list of other resources in his bibliography:

Codd, E. F. *Normalized Database Structure: A brief Tutorial. Proceedings of ACM SIGFIDET Workshop in Data Description, Access, and Control*, 1971.

Fleming, Candace C., and Barbara von Halle. *The Handbook of Relational Database Design*, Addison-Wesley Publishing Co. Inc., 1989.

Database Programming and Design. Miller Freeman Inc., 600 Harrison St. San Francisco, CA.

Individuals wishing to learn more about DataEase can contact the corporate office at (203) 374-8000.



GET THE LOWEST OS/2 PRICES-KWIK

VisualAge C++ for OS/2 version 3.0

by IBM

Version 3 of VisualAge C++ offers you visual programming tools that make building applications simpler.

3.5" disk with documentation	\$429.00 (MSR \$525.00)	Order number 448
CD ROM	\$370.00 (MSR \$449.00)	Order number 449
CD ROM with documentation	\$409.00 (MSR \$489.00)	Order number 450

call for prices on upgrades from v2/v2.1

VX•REXX for OS/2

by Watcom

Powerful and easy-to-use integrated environment for developing OS/2 2.x PM applications, including a project management facility, visual designer and debugger. New to this version are Notebooks, Containers, Sliders, Popup menus, DDE, objects and more.

\$95.00 (MSR \$99.00)

Order number 211

SPF/PC v 4.0

by Command Technology

SPF/PC is a powerful file manager and full-screen text editor that emulates IBM's mainframe ISPF/PDF, providing a familiar environment for mainframe programmers faced with the challenge of developing on a PC.

\$189.00 (MSR \$295.00)

Order number 186



These products and many more PLUS our lowest price guarantee.

OS/2 Express promises to match or beat nationally advertised prices for OS/2 software we carry. Just place your next order with OS/2 Express. If you find a vendor that can fill your order for less elsewhere (including shipping, handling, credit card fees and other surcharges), send or fax us the ad within seven days. We'll match it.

OS/2 EXPRESS
1-800-OS2-KWIK

For orders call 1-800-OS2-KWIK (1-800-672-5945) • Fax (612) 823-6267 • International (612) 823-6255

Circle Reader Service Number 21

OS/2 DEVELOPER BACK ISSUES

LIMITED NUMBER OF ISSUES AVAILABLE

The following Back issues of OS/2 Developer are available for \$9.95 each.
(Add \$2.50 for shipping)

YES, PLEASE SEND ME:

- | | |
|---|---|
| ☐ Winter 1990 | ☒ Fall 1992 SOLD OUT |
| ☐ Fall 1990 | ☐ Winter 1993 |
| ☐ Fall 1991 | ☒ Spring 1993 SOLD OUT |
| ☒ Winter 1992 SOLD OUT | ☐ July/August 1993 |
| ☐ Spring 1992 | ☐ Sept/Oct 1993 |
| ☒ Summer 1992 SOLD OUT | ☐ Nov/Dec 1993 |

Subscribe to OS/2 Developer for one year and save 33% off the cover price.

YES, PLEASE SEND ME:

- ☐ 1 year (six issues)—\$39.95 in the U.S.

Canada, Mexico, and international surface mail—add \$16 per year

International air mail—add \$30 per year

Mail this page to: **OS/2 DEVELOPER**

P.O. Box 1079

Skokie, IL 60076-9772

fax: 415-905-2233 phone: 800-926-8672

Please Print:

NAME

TITLE

COMPANY

ADDRESS

CITY

STATE

ZIP

☐ Check enclosed

☐ Charge my credit card:

☐ Visa

☐ Mastercard

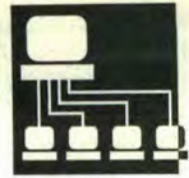
☐ American Express

CARD NUMBER

EXPIRATION DATE

SIGNATURE

BACKAD



Writing an OS/2 database application, using a genealogy database as an example.

By JAKE JACOB

A Multi-Platform Database Library

About 10 years ago I was looking through some family scrapbooks and photo albums when Mom said, "Why, you could put this all on your computer, couldn't you?" And without thinking (obviously!) I replied, "Sure, that's what they're for." Envisioning perhaps as many as 200 names, I started writing a simple program to record and organize the information and print the results.

By now my records include nearly 3,000 names, 5 MBytes of text data, and 150 Mbytes of photographs. It's time to start thinking database program! So when Simple Software Solutions offered me an opportunity to beta test its new database library, I asked some questions and was persuaded to try it. Although I don't expect to use all of its capacity, I wanted to be able to write my own data entry and reporting functions.

My genealogy database uses multiple tables to define parent-child and marriage relationships and link them to text and photographic data. When I finish all the "bells and whistles," users will be able to view and print information about an individual or a family. It's fun to step back from parent to grandparent and see their pictures and read about their activities.

AN EVENT-DRIVEN DATABASE LIBRARY: E_DB

This article describes e_Db, a database library product that will soon be available for most popular development platforms, including OS/2 Warp. The "e" in e_Db stands for events, since the database engine posts events back to the application to inform it of changes, additions, and deletions.

The product is a comprehensive set of functions, which allows a developer to create and maintain databases from applications written in C, C++, VisualBasic, and other languages that can access functions in a DLL.

Besides the basic INSERT, DELETE, UPDATE, and SELECT operations, e_Db provides many other useful functions.

The appealing aspects of e_Db are:

- It completely hides the details of opening, closing, and sequencing table and index files.
- Its search engine examines the search expression to determine what indices (if any) to use and how to conduct the search most efficiently.
- It posts events to inform the calling application of database changes, table additions, and indices dropped. These events are also used to allow customization of the database engine's behavior. For example, once the maximum number of locking retries has been reached, e_Db posts an event asking for direction: continue trying? stop?
- e_Db is available for OS/2 Warp, Windows 3.1, Windows NT, VisualBasic, and DOS. Versions for OS/2 for the PowerPC and Windows 95 will be available when these systems ship.
- The architecture defines a relationship between the database engine and its database drivers. A database driver DLL relieves the engine from having to know the details of file format. Version 1.0 of e_Db is shipped with a DBASE III database driver but other drivers can be developed. The functionality of the APIs remains the same regardless of the database driver being used.
- Data files created and maintained on one platform can be accessed by implementations on other platforms.

If you develop software in C or C++ under OS/2, Windows, NT, or DOS, and want to avoid complex database systems, this product is worth considering. e_Db's API set provides a consistent interface across all supported platforms. It offers a level of refine-



ment not found in most other database libraries. You never “see” a table or index file, just as you never “see” a file in more complex database systems. You don’t need to write code to scan tables or serialize access to tables.

On the other hand, `e_Db` lacks the user/password validation, referential integrity checks, and dedicated backup and recovery features found in larger, more complex, database systems.

FEATURES SUMMARY

With `e_Db` one can do the following:

- Create a multiple database dynamically
- Create and drop tables dynamically
- Manipulate tables in various file formats
- Handle multiple databases concurrently
- Create and drop table indices dynamically
- Submit SELECT statements against any table
- Submit multicolumn UPDATE requests
- Submit INSERT and DELETE requests
- Supply full multithreaded support under OS/2 and Windows NT
- Use wildcard characters ? and * to search the tables
- Set case sensitivity at the database level
- Compose complex, parenthesized expressions for SELECT, UPDATE, and DELETE statements
- Use C language structures to manipulate table rows
- Import and export data to/from a table with a single function call
- Catalog and uncatalog tables dynamically
- Notify when new tables or indices are created by other users
- Notify when maximum number of rows is reached during SELECT operations
- Notify application when tables are removed
- Open a database for exclusive, shared, or view only access
- Treat select results as an array of rows
- Supply a callback mechanism for SELECT statements
- Supply a callback mechanism for UPDATE statements that allows you to perform special updates.

DATA TYPES

Data types that are not directly supported by the chosen file format are emulated by its database driver. For example, in Table 1, `CT_WORD` (unsigned

Symbol

Description

<code>CT_FSTRING</code>	Fixed-length character
<code>CT_NUMERIC</code>	Formatted numeric value
<code>CT_BOOLEAN</code>	Boolean variable ('T' or 'F')
<code>CT_DATE</code>	Date column (YYYYMMDD)
<code>CT_VSTRING</code>	Variable length character (DBASE memo field)
<code>CT_BYTE</code>	Unsigned binary byte
<code>CT_WORD</code>	Unsigned short (two bytes)
<code>CT_DWORD</code>	Unsigned long (four bytes)
<code>CT_FLOAT</code>	Floating point number (double)
<code>CT_SHORT</code>	Signed short (two bytes)
<code>CT_LONG</code>	Signed long (four bytes)
<code>CT_TIMESTAMP</code>	Unsigned short (two bytes)
<code>CT_COUNTER</code>	Unsigned long (four bytes) (increments automatically)

Table 1. `e_Db` supported data types.

Column type

Description

<code>fstring(n)</code>	Fixed string of n characters
<code>numeric(w:d)</code>	Numeric value of 'w' bytes with 'd' decimal places
<code>boolean</code>	1-byte wide. Y or N.
<code>date</code>	8-byte wide date. YYYYMMDD
<code>vstring</code>	4-byte wide reference to a variable length string
<code>byte</code>	1-byte wide unsigned value
<code>word</code>	2-byte wide unsigned short value
<code>dword</code>	4-byte wide unsigned long value
<code>float</code>	8-byte wide double value
<code>short</code>	2-byte wide signed short value
<code>long</code>	4-byte wide signed long value
<code>counter</code>	4-byte wide unsigned long value
<code>timestamp</code>	2-byte wide unsigned short value

Table 2. `e_Db` column types.

short) is not directly supported in the DBASE III file format but the database driver emulates it using DBASE III's 'numeric' type. Regardless of the database driver you're using, your application always handles a `CT_WORD` as a two-byte unsigned short variable.

USING A DATABASE

Let's look at a few of `e_Db`'s APIs with some “snippets” of pseudo code to get a better idea of how they are used.

Before you call any other functions, you must create a database engine instance with a call to `DbmCreateEngineInstance`. This function returns a handle, which is used to create and/or open the database(s).

After the database is created, you can dynamically create and drop the tables and indices you need. The database instance handle is also used to “shutdown” the engine with `DbmShutdownEngineInstance`. The sequence involved in creating and manipulating a database is:

```
DbmCreateEngineInstance()
DbmCreateDatabase() // if it doesn't exist
DbmOpenDatabase()
DbmCreateTable()
DbmCreateIndex()
DbmInsert()
DbmSelect()
DbmUpdate()
...
...
DbmCloseDatabase()
DbmShutdownEngineInstance()
```

When you create a database the engine produces a database dictionary file named after the database name you specify. This file has the extension `.dbd`.

```
For example,
int rc;
HDBINST hdbinst;
rc = DbmCreateEngineInstance(&hdbinst);
if (rc == DBSUCCESS)
{
    rc = DbmCreateDatabase(
```




```

HEXP hexp;           // Expression handle
HTABLE hTable;       // Table handle

rc = DbmCreateExpression( // Create new (empty) expression
    hTable,           // The table handle
    &hexp)            // Expression handle returned here

```

Now add the appropriate expression terms.

```

rc = DbmAddExpressionTerm( // Add expression term
    hexp,                 // Handle to expression
    "Date_of_Birth",      // The column to compare
    OP_GE,                // (greater than or equal)
    "18000101");          // The date to compare

```

followed by,

```

rc = DbmAddExpressionOperation(
    hexp,                 // Handle to expression
    LOGOP_AND);          // Logical operation (and)

rc = DbmAddExpressionTerm( // Add expression term
    hexp,                 // Handle to expression
    "Date_of_Birth",      // The column to compare
    OP_LE,                // (Less than or equal)
    "18491231");          // The date to compare

```

At this point you're ready to make the call to DbmSelect.

```

HSELECT hselect;        // The select handle

rc = DbmSelect(          // Select rows from table
    hTable,              // The table handle
    "*",                 // Return all columns
    hexp,                // Selection criteria
    0,                   // Handle to join description (none)
    FALSE,               // No distinct rows
    1000,                // Limit returned rows to 1000 rows
    &hselect,            // Handle to selection
    0);                  // User defined data (not used here)

```

DbmSelect returns a handle to the selected results. Next, you need to determine if any rows matched the criteria you specified in the search criteria expression hexp:

```

long count;              // Number of rows in select results

rc = DbmQuerySelectCount( // Obtain number of rows selected
    hselect,              // Handle to select results
    &count);              // The count of rows found.

```

And now let's examine the data that was returned:

```

long i;                  // loop variable

for (i = 0; i < count && rc == DBSUCCESS; i++)
{
    // Extract the entire row of data into a structure

    rc = DbmGetSelectRowData( // transfer data to structure
        hselect,             // handle to select results
        i,                   // the row offset (0-based)
        (char *) &Person,    // the structure to fill in
        sizeof(Person),       // the size of the structure
    );
}

```

```

hdbinst,                // Database instance handle
"C:\\\GENEALOG",         // Produces C:\GENEALOG.DBD file
"My relations database", // Database description (pun
                        // intended)
"dbase3",               // Database driver (DBASE III)
"db3dll0",              // The driver's dll name (OS/2)
CRDB_OVERWRITE);        // Overwrite if exists

}

```

OPENING A DATABASE

You must open the database in one of its three modes before you can use it. In exclusive mode, you have full access to the database and can create and drop tables and indices. In shared mode, you cannot drop database objects, and in view mode, you are only allowed to search the tables, and no changes are permitted.

CREATING A TABLE

A table is defined by a column specification containing one or more column descriptors of the form "Name type(width)". See Table 2 for column types.

In this example,

```

HTABLE hTable;
int rc;
rc = DbmCreateTable(
    hdb,
    "Genealogy Main Table",
    "Person_Id counter",
    "LastName fstring(30)",
    "FirstName fstring(30)",
    "Date_of_Birth date",
    "Place_of_Birth fstring(50)",
    "Date_of_Death date",
    "Place_of_Death fstring(50)",
    "UpdateDate date",
    ...
    ...
    "Comments vstring",
    &hTable);

```

would create a table for use in a genealogy program application.

CREATING AN INDEX

Create an index with the DbmCreateIndex function. If an index already exists for the specified column, it will be rebuilt.

The database engine automatically updates the index as you add, delete, and modify table rows. It also decides which indices to use during the execution of a SELECT, UPDATE, or DELETE operation.


```

TRUE);    // trim trailing blanks

// OR, extract each column individually with DbmGetSelectColumn

rc = DbmGetSelectColumn(
    hselect,    // handle to select results
    i,         // the row offset (0-based)
    0,         // Table handle (not used here)
    "Date_of_Birth", // Column to extract data from
    szDateofBirth, // Store the extracted data here
    TRUE);    // Trim blanks from result
}

```

Finally, free the memory associated with the select results handle

```
DbmFreeSelect(hselect);
```

Search expressions are reusable from one operation to another, however, they can only be used by one thread at a time.

Listing 1. An example of search criteria. (Continued from page 59.)

ADDING ROWS TO THE TABLE

A "data space" is associated with a particular table and is an internal representation of the data buffer that holds the data for one row. Once you have a valid "data space handle," you can set the column values or, if you have an appropriate C structure you can transfer its contents to the data space all at once. A data space is created by a call to `DbmCreatDS`.

The easiest way to insert a row into a table is by transferring the contents of a structure (or VisualBasic user-defined data type) directly to a data space before calling `DbmInsert`. If you are not using structures to manipulate the table rows, you must fill the data space by using `DbmSetColumn` for each column.

SEARCHING A TABLE

To search a table, call function

```

HTABLE hTable
int rc;
long count;
HUB hub;

rc = DbmCreateExpression(hTable, &hexp);
if (rc == DBSUCCESS)
{
    DbmAddExpressionTerm(hexp, "LastName", OP_EQ, "Smith");
    DbmAddLogicalOperator(hexp, LOGOP_AND);
    DbmAddExpressionTerm(hexp, "FirstName", OP_EQ, "John");

    rc = DbmCreateUpdateBlk(
        hTable,    // Table handle
        &hub);    // Update block handle returned here

    if (rc == DBSUCCESS)
    {
        // Set John Smith's date of birth to 10-12-53

        DbmAddUpdateAction(hub, "Date_of_Birth", OP_SET, "19531012");

        rc = DbmQuickUpdate(
            hexp,    // Handle to search expression
            hub,    // Update block handle
            &count); // Count of updated rows returned here
            // Only one row in this case.
        DbmFreeUpdateBlk(hub);
    }
    DbmFreeExpression(hexp);
}

```

Listing 2. How to update database reflecting correct date of birth.

`DbmSelect`. For example, you may want to search the "Genealogy Main Table" for all people born between 1800 and 1850. The search criteria is put together in the form of an expression before you call `DbmSelect`. An example is shown in Listing 1.

UPDATING THE TABLE ROWS

You can choose to update one or more columns in each call to `DbmUpdate`. This function uses a search expression like the one used by `DbmSelect`.

You can update a column by setting it to a specific value, adding to its current value, or subtracting from its current value. You can also perform search/replace on the value of a string column; add or subtract days, months, and/or years to a date column; or change the case of a column to upper or lower case.

If an update procedure is specified, the engine invokes it before updating a column's value. The update procedure can choose to continue, make its own changes to the column, or bypass the operation completely. This mechanism allows the programmer to perform update actions that are not directly supported by the database engine.

The special form of `UPDATE_ADD` and `UPDATE_SUB` allows me to add or subtract days, months, and years to or from a column of type `CT_DATE`.

Listing 2 is an example of how to update my database to reflect John Smith's correct date of birth.

CATALOGUING A TABLE

If you have a table in the proper file format (dbASE III for version 1.0), then you can incorporate it into the database by "cataloguing" it with `DbmCatalogTable`. On return from this function, you can use the table as if you had created and populated it yourself. Use function `DbmUncatalogTable` to perform the reverse operation.

EVENT HANDLING

The `e_Db` database engine optionally calls back your application when certain events occur. Your event handler receives a message ID that tells it which event took place. Event messages are shown in Listing 3.

IMPORTING AND EXPORTING TEXT

You can import text from a file into an existing table using `DbmImportTable`. The file must consist of lines of text separated by CR-LF combinations. The first line of the file must be a table definition string similar to the one passed to `DbmCreateTable`. For example,



Here is a list of e_Db functions by category:

Instance handling functions:

DbmCreateEngineInstance
DbmShutdownEngineInstance

Database handling functions:

DbmQueryDbDrvClass
DbmOpenDatabase
DbmCloseDatabase
DbmCreateDatabase
DbmDropDatabase
DbmQueryDbInfo
DbmQueryDbTables
DbmQueryDbIndices
DbmPromoteAccess
DbmSetCaseSensitivity
DbmSetRetryParameters

Table handling functions:

DbmGetTableHandle
DbmCatalogTable
DbmCreateTable
DbmDropTable
DbmExportTable
DbmImportTable
DbmQueryTableInfo
DbmQueryTableColumns
DbmQueryTableIndices
DbmUncatalogTable
DbmWidestTableColumn

DbmPackTable
DbmGetTableRowCount

Search criteria expression handling functions:

DbmCreateExpression
DbmAddExpressionTerm
DbmAddStringTerm
DbmAddExpressionParen
DbmAddLogicalOperator
DbmFreeExpression

Join handling functions:

DbmCreateJoin
DbmAddJoinTerm
DbmFreeJoin

UPDATE handling functions:

DbmCreateUpdateBlk
DbmAddUpdateAction
DbmAddUpdateString
DbmUpdate
DbmQuickUpdate
DbmFreeUpdateBlk
DbmAlloc
DbmFree

INSERT handling functions:

DbmCreateDS
DbmClearDS
DbmGetColumnWidth
DbmSetColumn
DbmSetExtColumn
DbmSetStruct

DbmQuickInsert
DbmInsert
DbmGetColumn
DbmFreeDS

Index handling functions:

DbmCreateIndex
DbmDropIndex

SELECT handling functions:

DbmSelect
DbmQuickSelect
DbmFreeSelect
DbmFreeSelectRow
DbmQuerySelectCount
DbmGetSelectStruct
DbmGetSelectColumn

DELETE handling functions:

DbmDelete
DbmQuickDelete

Event handling functions:

DbmSetEventProc
DbmDefaultEventPro

Special performance-related functions:

DbmLockTable
DbmUnlockTable

Error handling function

DbmRetrieveString

name fstring(10), address fstring(30) CR-LF
"John Doe", "2659 SE 14th Street" CR-LF

To export the contents of a table to a text file use DbmExportTable.

LOCKING & UNLOCKING TABLES

Locking and unlocking of tables is handled by the database engine. Whenever needed, calls to DbmInsert, DbmSelect, DbmDelete, DbmUpdate, lock the table(s), perform the given operation, and then unlock the table(s).

UTILITIES

Utilities are included to allow you to

quickly create databases, tables, indices, query table contents, update columns, and catalog tables. I found GENHEAD.EXE a particularly useful utility. When given the name of a database dictionary file (.dbd), it produces a header file with all the typedef statements required to manipulate the rows in native C or C++. Another utility called GENVB.EXE generates a similar file for use with VisualBasic.

CLOSING THOUGHTS

I found e_Db a good compromise between a simple database library package and a complex database sys-

tem. It is a database library that does most of the work for me while providing portability, speed, ease of use, and robustness.

Jake Jacob is a retired (really!) IBMer who began programming when it was called wiring control panels. He's been involved with PCs since their inception and has taught programming languages, desktop publishing, graphic arts and word processing classes. He is currently an independent consultant working in desktop publishing. He can be contacted at (407) 395-6069 or at CompuServe 76305,2510.

Event symbol Description

DBM_NEW_TABLE	Another user added a table to the database
DBM_NEW_INDEX	Another user added an index to a table
DBM_RETRY_APPROVE	If the internal number of retries is exceeded, the system calls your event handler to decide whether to continue or to terminate the operation.
DBM_SELECT_LIMIT	Whenever a select operation encounters more rows than you specified as maximum, this event is sent to your event handler so that it can decide how to proceed. DBM_SELECT_ROW. Optionally you can receive a message for each row matched in a call to DbmSelect.

Table 3. Event message table.

REFERENCES

The e_Db product is shipped with three manuals: an installation guide, programmers' guide, and a programmers reference. A technical document providing additional product information is also available.

All are available from: Simple Software Solutions Inc., 2659 SE 14 St., Pompano Beach, Fla. 33062. Voice: 305-782-8385, fax: 305-782-0818. CompuServe 72143,2024, internet cportela@gate.net



The staff of OS/2 Developer put together this special section to guide you through the various software tools specifically for the PowerPC. The products described in this guide are based on surveys sent to vendors; they are provided as a service to you and are free to vendors. These listings do not represent an endorsement by OS/2 Developer. Although every effort has been made to ensure accuracy, we do not assume any responsibility for error or omission in this section. Contact vendor for availability.

Tools for the PowerPC Buyers Guide

ABRAXAS SOFTWARE INC. *Circle No. 130*

PCYACC 5.0 is a complete Language Development Environment that generates ANSI C and C++ source code from input Language Description Grammars for building assemblers, compilers, interpreters, browsers, and so on. This product generates C++ and YACC classes for OS/2 2.0 Presentation Manager. In addition, PCYACC builds systems with full error handling, recovery, and reporting. Price: \$995.

CodeCheck 5.0 is a programmable tool for managing all C and C++ source code on a file or project basis. It is input-compatible with all variants of K&R, ANSI C, and C++. CodeCheck allows corporate coding and project specification standards to be automated for compliance validation and will identify code that will not port between platforms. Price: \$995.

Abraxas Software Inc., 5530 SW Kelly, Portland, Ore. 97201, (503) 244-5253, fax (503) 244-8375.

ACCESS *Circle No. 131*

dSIGN for Windows 2.5 databased publishing package allows retailers and distributors to create shelf signs, bar code labels, and customer mailings with WYSIWYG, linking them to databases using FoxPro 2.5 Data Manager and downloading to Portable dSIGN/2-32 in-store under OS/2. The product allows printing using PostScript laser printers. Price: \$995.

Access, 3131 Western Ave., Ste. 530, Seattle, Wash. 98121-1028, (206) 285-4994, fax (206) 285-1807.

ACCUSOFT CORP. *Circle No. 132*

AccuSoft Image Format Library 5.0 supports raster formats: TIFF, JPEG, PCX, GIF, DIB, WPG, BMP, TGA, PICT, EPS, WMF, DCX, PCD, IFF, SUN, PSD, ASCII, and 19 more.

The product has the ability to import, export, convert, compress, scan, display, and print. Image processing includes: rotate, invert, zoom, pan, scroll, color reduction, edge detect, laplacian, thumbnails, and so on. It works with any OS/2 language that supports DLLs. Price: \$1,295.

Accusoft Corp., 2 Westborough Business Pk., Westborough, Mass. 01581, (508) 898-2770, fax (508) 898-9662.

ARITY CORP. *Circle No. 133*

Prolog 32 for OS/2 provides power of the Arity/Prolog language in addition to the superior performance of the 32-bit environment. This product eliminates many imposed 16-bit environment limitations for language and system performances. Price: \$995.

Arity Corp., Damon Mill Sq., Concord, Mass. 01742, (508) 371-1243, fax (508) 371-1487.

AUTOTESTER INC. *Circle No. 134*

AutoTester for OS/2 3.0 is an automated testing tool designed specifically for OS/2 PM applications. AutoTester builds structured, object-oriented, well-documented tests as you operate an application. The product supports variables and external test case data, conditional branching, and text and bitmap data comparisons. In addition, it includes a menu-driven interface as well as a powerful command language for advanced scripting needs. Price: contact vendor

AutoTester Inc., 8150 North Central Expwy., Ste. 1300, Dallas, Tex. 75206, (214) 368-1196, fax (214) 363-2886.

BENTLY SYSTEMS INC. *Circle No. 135*

MicroStation 5.5 is a full-featured 2-D/3-D professional computer-aided design product. In addition to its consistent, intuitive, graph-

ical user interface and flexible command structure, the 2-D production drafting tools are the force behind MicroStation's work-group-oriented approach. It is capable of running programs written in the MicroStation development language. Price: \$3,950.

Bently Systems Inc., 690 Pennsylvania Dr., Exton, Pa. 19341, (610) 458-5000, fax (610) 458-1060.

CAP AUTOMATION

Circle No. 136

CAP Accountant is a DOS-based accounting system that can be run on a cash or accrual basis and will generate real-time financial and accounting reports. In addition to doing regular general ledger accounting functions, it will also keep track of your checkbook balance and allow easy bank reconciliation. Price: \$495.

CAP SellWise 95.2 replaces manual work with automatic inventory control, product re-ordering, serial number tracking, accounts receivable, and customer tracking. Sales stations continue to operate following any network failure. Information appears instantly with the standard reports for any date range without posting, closing periods, batch processing, or manual calculating. SellWise captures the information for each sale, inventory, customer, and vendor, and shares it with back-office management in real time. Price: \$1,495 for a single user.

CAP Automation, 3737 Ramona, Fort Worth, Tex. 76116, (800) 826-5009, fax (817) 560-8249.

CIRRUS TECHNOLOGY INC.

Circle No. 137

UniteLite 1.5 is designed for small workgroups to reduce storage and handling requirements associated with paper files and folders. It is an image and photograph storage and retrieval tool based on the OS/2 Workplace Shell and OS/2 Warp. UniteLite's capabilities include scanning, viewing, finding, and printing forms, reports, and other key documents. Price: \$295.

Unite CD-Maker 1.3 uses IBM's SOM and Workplace Shell technology for creating custom CDs. This product takes advantage of OS/2's multithreaded performance to increase the speed and reliability of CD-ROM creation. Two CD recording methods are supported: "bum-on-the-fly" or build a virtual CD image file. Both methods produce CD-ROMs that conform to the ISO 9660 format and can be read from any OS/2 system.

Price: \$249.

Cirrus Technology Inc., 5301 Buckeyes-town Pike, Frederick, Md. 21701, (301) 698-1900, fax (301) 698-1909.

COMPUWARE CORP.

Circle No. 138

Uniface is a robust development environment, which is open and portable. Applications built with Uniface have simultaneous read/write access to numerous database and file management systems, including DB2, Oracle, Sybase, Informix, INGRES, and others. A single application runs under multiple GUIs and character modes without any additional programming. Regardless of interface, Uniface applications exploit each environment. This product also works with several CASE tools, including those from Knowledgeware and Intersolv. Uniface has been used to build enterprise client/server software applications in a wide range of industries including financial services, pharmaceuticals, telecommunications, government, and manufacturing. Price: contact vendor.

Compuware Corp., 31440 Northwestern Hwy., Farmington Hills, Mich. 48334-2564, (800) 356-3608, fax (810) 737-7119.

CONTRACCT SYSTEMS CORP.

Circle No. 139

ContrAcct 3.1, a fully integrated contractor accounting/job cost management system, is designed for contractors. ContrAcct includes accounts payable, accounts receivable, cash management, equipment, general ledger, inventory, job cost, payroll, purchase order, and report writer. It is a single entry system of source documents with all internal postings performed automatically. In addition, this product supports multiple companies and multiple users, and offers custom programming, individual professional training, telephone and modem support, and annual updates. Price: contact vendor.

ContrAcct Systems Corp., 208 N. Washington St., Naperville, Ill. 60540, (800) 562-2678, fax (708) 355-8675.

COPIA INTERNATIONAL

Circle No. 140

Accsys for Paradox 6.0 enables programmers to create and/or update Paradox files. It is an interface between C or VisualBasic and Paradox (through 5.0), providing the programmer control over Paradox table files, primary, and secondary index files. Accsys provides a portable multiplatform database system with a high-level and low-level API

for faster application development. Price: \$495.

AccSys for xBase 6.0 enables programmers to create and/or update files of dBASE III, IV, Clipper, and FoxPro. It combines the power of dBASE with the efficiency of C or VisualBasic to create more efficient applications. AccSys for xBASE provides a portable multiplatform database system with a high- and low-level API for faster application development. Price: \$495.

Copia International, 1342 Avalon Ct., Wheaton, Ill. 60187, (800) 689-8878, (708) 665-9841.

CYGNET PUBLISHING

TECHNOLOGIES INC.

Circle No. 141

Scribe Document Production Software 7 (1750) is cost-effective publishing software designed for the production of large and complex documents. Producing high-quality output, it integrates text, graphics, tables, and external data from database management systems. Price: \$895.

Cygnat Publishing Technology Inc., 355 5th Ave., Ste. 1515, Pittsburgh, Pa. 15222, (412) 471-2070, fax (412) 391-7215.

DST BELVEDERE

Circle No. 142

The Global Portfolio System (GPS) 3.0 is a UNIX-based multicurrency portfolio accounting and performance measurement solution for institutional investment advisors. GPS supports all transactions and provides complete accounting for all securities and currencies. With its rules-based design, GPS offers the extensibility to meet the demands of new markets as they emerge. GPS is a client/server solution that has a platform, relational database, and is graphical user interface independent. Price: contact vendor.

DST Belvedere, 470 Atlantic Ave., Boston Mass. 02210, (617) 482-8800, fax (617) 482-8878.

FAIRCOM CORP.

Circle No. 143

d-tree Development Tool 3.4A offers beginning programmers a bridge to the control, flexibility, and portability of C development. Experienced programmers increase results while maintaining low-level flexibility. Features include, productive front-end for c-tree, complete portable screen handler, menu/HELP management, data dictionary, file-/field-level editing/validation, resource swapping, and run time control of program resources. Price: \$495-\$745.

FairCom Server 6.04.24 is a high-per-

formance, multithreaded, transaction processing server. Features include, industrial quality transaction processing, full commit and rollback, intermediate save points and complete logging, automatic log management, restart/disaster recovery, user passwords, access security and on-line administration, deadlock detection/resolution, and read/write locks at the record/individual key level. Price: \$295-\$3,795.

FairCom Corp., 4006 W. Broadway, Columbia, Mo. 65203-0100, (314) 445-6833, fax (314) 445-9698.

FELSINA SOFTWARE

Circle No. 144

A-Talk 1.6 is the Microsoft Office Compatible telecommunications package, which includes, Microsoft Mail, cc:Mail, Notes Mail, DDE and Net-DDE, VT220, VT100, VT52, Tektronix 4014, H19, VIDTEX, ANSI, TTY; X/Y/ZModem, Kermit, CompuServe B+, ASCII; BBS, and chat mode, drag-and-drop, history buffer, phone books, clipboard, help, over 280 modems, script language, editor, and autoscript. Microsoft Office's compatible features include toolbar, tooltips, statusbar, and advanced help. A-Talk is a certified OS/2 Warp compatible by NSTL. Price: \$79.95.

Felsina Software, 4440 Finley Ave., Ste. 108, Los Angeles, Calif. 90027, (213) 669-1497, fax (213) 669-1893.

FIRST BYTE INC.

Circle No. 145

ProVoice Tool Kit 2.1 allows programmers to add synthesized speech to their applications. The program passes text strings to a speech driver, while translating the text into audible speech. All of the necessary tools and examples are provided to manipulate the ProVoice speech technology. Price: \$599.

First Byte Inc., 19840 Pioneer Ave., Torrance, Calif. 90503, (310) 793-0610, fax (310) 793-0601.

FOOTPRINT

SOFTWARE INC.

Circle No. 146

IBM Works 2.0 is currently included as a part of the OS/2 Warp BonusPak. It is an integrated productivity package offering word processing, spread sheets, charting, database, report writer, and personal information management tools. IBM Works provides integration across the modules and with the Workplace Shell. Price: contact vendor.

Footprint Software Inc., 53 Yonge St., Toronto, Ont., Canada. M5E1J3,

ADVERTISER INDEX

Reader Service	Company Name	Page
22	American Coders	70
23	Chipchat-Cawthon	70
18	Cirrus	52
14	ColoradoOS/2	33
32	Crossley	71
19	DB Technologies	53
12	Development Technologies	30
24	Essex	70
2	Hockware	1
13	Indelible Blue	31
-	Inmark	COV4
5	JBA	14
31	Jagre	71
25	Keller Group	70
20	Lieberman	53
8	MHR	19
1	Metaware	COV2
16	MicroEdge	43
-	Microwork	28
10	Mitnor	24
9	On-Line Data	19
26	Opt-Tech	70
21	OS/2 Express	55
27	Perez	70
3	Programmer's Paradise	2
-	Quadron	25
4	Rimstar	4
6	Segue	17
11	Soffront	71
17	SoftBridge	52
29	Softel	71
33	Spectrum	71
15	Suitable Alternatives	36
7	Watcom	18
34	Zinc	COV3

(416) 860-0477, fax (416) 860-1780.

IBM CANADA

Circle No. 147

VisualAge C++ for OS/2 Warp 3.0 extends IBM's programming tools to the PowerPC. The tools, including parts-based Visual and Data Access Builders, IBM Open Class libraries, integrated C++ programming tools, and robust Direct-To-SOM C/C++ compilers produce optimized code and are consistent with VisualAge C++ for OS/2 and the entire VisualAge C++ family. PowerPC code generation support, in an upcoming version of VisualAge C++ for OS/2, will also allow development of both Intel and PowerPC applications from a single development platform. Price: contact vendor.

IBM Canada, 1150 Eglinton Ave., North York, Ont., Canada. M3C1V7, (416) 448-2647, fax (416) 448-4414.

IBM

Circle No. 148

The Developer Connection for OS/2 subscription can increase productivity by providing the most updated tools, pre-release software, product demos, and information needed on today's medium CDs. An annual subscription to *The Developer Connection News* keeps tools and information from becoming obsolete. Each year, receive four volumes including CDs and *The Developer Connection News*, each packed with the tools and information you need for your Intel and PowerPC OS/2 development efforts. Price: \$199.

The IBM Developer's Toolkit for OS/2 Warp Connect (PowerPC Edition) will be available as part of The Developer Connection for OS/2. The toolkit is positioned to assist developers of the Power Series family in specific applications and allow a recompile of existing OS/2 applications running on x86 processors to run on the IBM Power Series family. The toolkit includes, tools for building programs such as linker, archiver, resource compiler, object and executable file analysis tools; resource editors and build tools; SOM compiler and emitters; headers and import libraries; online documentation; code tutorials; and debugging aids. Price: \$199.

IBM, 100 N.W. 51st. St., Boca Raton, Fla. 33432, (407) 443-5180, fax (407) 443-2453.

ICVERIFY INC.

Circle No. 149

ICVerify 6.5 is a complete PC-compatible credit authorization software program designed to provide credit card authorization, check acceptance, and ATM/debit card authorizations with

menu-driven simplicity. ICVerify can operate by itself using a pop-up windows or can be integrated into other application programs. Additional features include: signature capture, multiple transactions/queuing, extensive reporting/statistics, address verification, modem sharing, and install-ment/recurring processing. Price: \$349-1,999.

ICVerify Inc., 473 Roland Wy., Oakland, Calif. 94621, (510) 553-7500, fax (510) 553-7553.

IMAGESTAT CORP.

Circle No. 150

ImageCare 2.3, a document management system, is designed with a flexible graphical interface. Using drag-and-drop techniques, the document management capabilities, including scanning, printing, faxing, and filing, can be accessed. This product fully supports the PowerPC under OS/2 as well as OS/2 LAN Sever and the IBM 3995 Optical Library family. Price: contact vendor.

ImageStat Corp., 2950 31st St., Ste. 220, Santa Monica, Calif. 90405, (310) 392-1100, fax (310) 392-6250.

INFINITYSYSTEMS CONSULTING

Circle No. 151

Infinity is a PC LAN-based policy and claims administration system, which fully integrates rating, policy issue, policy management, claims administration, client management, statistical reporting, and accounting utilizing IBM's OS/2 operating environment. Price: contact vendor.

Infinity Systems Consulting, 7500 Old Georgetown Rd., Ste. 700, Bethesda, Md. 20814, (212) 675-1757, fax (212) 675-5479.

KELLER GROUP INC.

Circle No. 152

Fax Client /Server API Toolkit 2.1 provides client/server fax services to applications on OS/2. Supported fax hardware includes fax modems, Mwave, SatisFAXtion, Brooktrout, GammaFax, and others. The API works with FaxWorks for OS/2, FaxPort for OS/2, and other OS/2 fax software products in any standalone, LAN, or multiline configuration. Price: \$995.

Fax Printer Driver Toolkit 2.1 facilitates writing applications that create and send fax documents. The printer driver can be used from REXX, DOS, Windows, 16-bit OS/2, and 32-bit OS/2 programs. The printer driver works with FaxWorks for OS/2 and FaxPort for OS/2 in any standalone, LAN, or multiline configuration. Price:

\$295.

Keller Group Inc., 8600 Jewel Ave. N., Stillwater, Minn. 55082-9351, (612) 429-7273, fax (612) 653-1987 or (800) 329-3293.

KNOWLEDGE

DYNAMICS CORP.

Circle No. 153

Winstall Multi-Platform 4.0 is an installation toolkit including distribution disk builder, full error handling capabilities, system file editor, one .EXE for both DOS and Windows, full C source code, no royalties, and a 30-day guarantee. Price: \$499.95.

Install Professional 3.3 is a professional installation toolkit for MS-DOS or OS/2. Install features include distribution disk builder, full error handling capabilities, system file editor, full C source code, no royalties, and a 30-day guarantee. Price: \$399.95.

Knowledge Dynamics Corp., P.O. Box 780068, San Antonio, Tex. 78278, (210) 979-9424, fax (210) 979-9004.

LOGOS SYSTEMS INC.

Circle No. 154

AmbuQual Ambulatory Quality Assessment and Quality Management System

6.1.1 focuses on issues that have the potential to impact the quality of healthcare offered to patients under the new health programs proposed by the executive and congressional branch of the U.S. Government. The software measures provider/staff performance; continuity of care; enhances the quality of patient care; minimizes patient risk; and measures patient compliance and organization performance. Price: contact vendor.

Logos Systems Inc., 8303 Southwest Fwy., Ste. 500, Houston, Tex. 77074, (713) 777-8229, fax (713) 777-9512.

MACROMEDIA

Circle No. 155

Director 4.0 for OS/2 facilitates the combining of text, graphics, animation, sound, digital video, and interactivity to create multimedia productions. Director will produce corporate presentation; entertainment; and educational CD-ROM titles, simulations and visualization; kiosks; digital publications; electronic reference materials; and software demo disks. Price: \$1,195.

Macromedia, 600 Townsend St., San Francisco, Calif. 94103, (415) 252-2000, fax (415) 626-0554.

MICROEDGE INC.

Circle No. 156

Visual SlickEdit 1.7 features CUA, Brief, Emacs, and VI emulations. This editor will facilitate programming with its project management, SmartPast, syn-



tax color-coding, and built-in dialog editor. There is a 30-day money-back guarantee. Price: \$295.

MicroEdge Inc., P.O. Box 18038, Raleigh, N.C. 27619-8038, (800) 934-EDIT, fax (919) 831-0101.

NETWORK

TELEMETRICS INC.

Circle No. 157

Proview-SNA 1.4 software monitors response time and availability parameters on complex, heterogeneous networks. It generates active traffic from LAN PC workstations to SNA hosts and network file servers. It then measures the end-to-end delay users are actually receiving across the network. Network-wide results are forwarded to a central management station where data is stored in a DB2/2 database. Charts and tabular reports of real-time and historical data are produced to verify service level agreements and identify performance trends. Price: \$25,000.

Network Telemetrics Inc., 30125 Agoura Hills, Calif. 91301, (818) 597-7900, fax (818) 597-7929.

ON-LINE DATA

Circle No. 158

OnCmd xBase for OS/2 harnesses the power and the speed of OS/2, enabling you to develop new or migrate existing applications from dBASE, FoxPro, or Clipper to native 32-bit GUI applications environment. It comes with screen painter, report writer, DLL/DDE support, 350+ functions/commands and is client/server ready. Unlimited run time and server add-on licensing available. Price: \$249.

On-Line Data, 5 Hill St., Kitchener, Ont, Canada. N263X4, (519) 579-3930, fax (519) 579-2130.

PEERLOGIC INC.

Circle No. 159

Pipes Platform 3.5 message-oriented middleware provides communications services for distributing applications across multiple operating systems, hardware platforms, and mixed networks. These services include, asynchronous, process-to-process messaging; dynamic name series; session management services; and cross-platform and multiprotocol support. Pipes Platform's flexible communications system facilitates developing, distributing, and maintaining multitier, distributed applications. Price: \$200.

PeerLogic Inc., 555 DeHaro St., San Francisco, Calif. 94107-2348, (800) PEER-601, fax (415) 626-4710.

PEGASUS DISK TECHNOLOGIES INC.

Circle No. 160

Pegasus Optical File System 2.3 is a transparent file and volume management system for optical drives and jukeboxes under IBM OS/2 2.x and OS/2 Warp. Optical drives and jukeboxes appear as standard storage devices. This product provides a common file system format on optical media that is transportable across operating-system platforms. The software tracks optical volumes stored online, nearline, or offline documents. Price: contact vendor.

Pegasus Disk Technologies Inc., 1600 S. Main St., Ste. 110, Walnut Creek, Calif. 94598, (510) 938-5340, fax (510) 938-5341.

PROCOL INC.

Circle No. 161

Procol RDBMP 4GL 6.0 is ideal for all phases of application development, as the product is designed to be portable, coherent, compact, and user-friendly. Procol's nonprocedural language enable 100% of development requirements and offers many advanced features. Price: starting at \$950.

Procol Inc., 5995 Gouin Blvd. W., Ste. 300, Montreal, Que, Canada. H4J 2P8. (514) 336-8551, fax (514) 336-8565.

QUANT TRADING INC.

Circle No. 162

Sector Trading System 2.0 is a real-time, fixed-income, front-office system used in trading treasury, municipal, sovereign and corporate debt markets. An extremely powerful and customizable object-oriented application, Sector provides real-time security pricing models, analytics, risk analysis, trade-entry, and P&L in an intuitive graphical-user environment. Price: contact vendor.

Quant Trading Inc., 100 Prospect St., Stamford, Conn. 06901, (203) 357-1919, fax (203) 6119.

RAIMA CORP.

Circle No. 163

Raima Database Manager 4.0 is a database engine for C and C++ application development providing a low-level C-API library with over 200 functions for database control. The combined relational and network database model features allow modeling of complex, real-world database designs with high-performance record access. Referential integrity is supported at the DDL level. It includes support for transaction tracking, logging, and automatic recovery. In addition, it supports all leading OS and computing environments in single- or multi-user configurations. Price: \$595 and up.

Raima Object Manager 3.0 is a class library that allows use of Velocis Database Server and Raima Database

Manager as an OODBMS. ROM encapsulates object storage and database navigation into C++ class definitions to provide a consistent object-oriented interface to Raima databases. The library adds persistence to any C++ object, object identity, class or type hierarchies, container classes, safe type checking, VLT, BLOB, Referential Integrity, server extensions, and so on. Price: \$495 and up.

Raima Corp., 1605 NW Sammamish Rd., Ste. 200, Issaquah, Wash. 98027, (206) 557-0200, fax (206) 557-5200.

RHINTEK COMPUTER ENGINEERING INC.

Circle No. 164

Rhino Com 1.5 is developed for OS/2's Presentation Manager. It is a general purpose, industrial strength communications package emphasizing reliability, versatility, and ease-of-use. RhinoCom is multithreaded, can handle multisessions, and contains multimedia sounds. This product also features advanced workplace Shell technology, REXX scripting, and macro capabilities. Price: \$199.

Rhintek Computer Engineering Inc., 8835-C Columbia 100 Pkwy., Columbia, Md. 21045, (410) 730-2575, fax (410) 730-5960.

SUNDIAL

SYSTEMS CORP.

Circle No. 165

Relish 2.2, time management software for workgroups and individuals, contains calendars, to dos, and a phone book that is integrated for reliability. In addition, this product includes desktop objects for common and customizable views, keyword searching, and drag-and-drop convenience. Relish runs on the PowerPC and mixed Intel-PowerPC environments. Price: contact vendor.

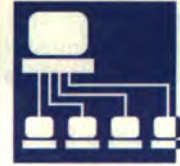
Sundial Systems Corp., 909 Electric Ave., Ste. 204, Seal Beach, Calif. 90740, (310) 596-5121, fax (310) 598-9851.

SPG INC.

Circle No. 166

ColorWorks for OS/2 PowerPC 1.0 is a 32-bit multitasking, multithreaded graphics paint image-processing, and image-editing software. ColorWorks eliminates the hourglass cursor, can save up to 90% of the memory usually needed to edit images, and can spread work across up to 64 processors-for workstation performance and beyond on a PC. Price: contact vendor.

SPG Inc., 155505 Bull Run Rd., Ste. 303, Miami Lakes, Fla. 33014, (305) 823-6602, fax (305) 823-2753.



The staff of OS/2 Developer put together this special section to guide you through the various software tools specifically for database development. The products described in this guide are based on surveys sent to vendors; they are provided as a service to you and are free to vendors. These listings do not represent an endorsement by OS/2 Developer. Although every effort has been made to ensure accuracy, we do not assume any responsibility for error or omission in this section.

Database Development Tools

Buyers Guide

ACUCOBOL INC.

Circle No. 101

Acu4GL provides seamless access to Informix, Oracle, and Sybase databases from new or existing applications written in AcuCOBOL-85 by generating SQL queries. With Acu4GL, users can benefit from RDBMS technology without learning SQL, rewriting applications, or retraining. Applications need to be compiled only once. This product eliminates the need to add embedded SQL to existing COBOL programs. Instead, industry-standard COBOL I/O verbs generate SQL to access the database. This bidirectional information exchange allows users to utilize fourth generation report writers and query tools while retaining the use of their COBOL application. Price: contact vendor.

Acucobol-85 is a portable COBOL compiler for microcomputers, minicomputers, and mainframes available. It produces compact, yet optimized machine-independent object code and runs on over 600 different platforms without recompiling. Acucobol-85 is source code compatible with RM/COBOL, VAX COBOL and DG-ICOBOL and includes its own user-replaceable file system. Price: contact vendor.

Acucobol Inc., 7950 Silverton Ave., Ste. 201, San Diego, Calif. 92426-6344, (619) 689-7220 or (800) COBOL85, fax (617) 566-3071.

APICAL SOFTWARE INC.

Circle No. 102

GpfRexx 1.0 is an OS/2 PM visual development tool that makes possible the use of REXX for implementation of custom logic. It is suitable for individuals and professional programmers, and offers all the power necessary to create full-functioned applications and/or working prototypes. There are no

royalties on GpfRexx generated applications. Its design files are compatible with Gpf 2.1, permitting designs or prototypes to be ported to C/C++ without having to recreate the GUI, GUI management logic, or Help. Price: \$199.

Gpf 2.1, is a visual GUI design environment that is not industry specific, which offers full support for OS/2 2.x and OS/2 1.3x, a CUA'91 compliant GUI, and Gpf generates C or C++ source as well as Help source and all ancillary files to build the design. Gpf automatically generates all the logic to manage the entire user interface, leaving the designer free to concentrate on custom application logic. Price: \$1,295.

Apical Software Inc., 40 Falls Rd., P.O. Box 432, Moodus, Conn. 06469-0432, (203) 873-3300, fax (203) 873-3302.

BACHMAN INFORMATION SYSTEMS INC.

Circle No. 103

Bachman/Analyst 4.2 is a comprehensive and scalable business-modeling solution for today's application development organizations. It supports a range of platform-independent modeling tasks, ranging from high-level business data and process modeling through detailed application modeling. The Bachman/Analyst enables companies to build flexible, in-depth models that accurately reflect business requirements and become long-term reusable assets for the organization as it grows and changes. Price: \$10,000 per seat.

Bachman/DBA for DB2 4.2 provides a graphical interface for rapid database design, describes database contents completely, and is fully integrated with tools for DB2 data-



base design. This product helps design and implement databases for a variety of other relational databases including Sybase and Microsoft SQL Server, Oracle, Informix, Ingres, Rdb, IBM's DB2/2, and OS/2 Extended Services Database Manager. Price: \$15,000.

Bachman Information Systems, 8 New England Executive Pk., Burlington, Mass. 01803, (617) 273-9003, fax (617) 229-9904.

BRIGHTWARE INC. Circle No. 104

ART*Enterprise for OS/2 1.0 R1 is a tool for building intelligent software applications that integrate databases, documents, and business policies. It provides business object modeling, business rules that automate corporate policies, and case-based retrieval (CBR) technology, which permits natural language retrieval of objects. A supported DBMS includes DB/2, Oracle, Sybase, SQL Server, and others via Q+E. ART*Enterprise applications are fully portable across Windows, OS/2, and Unix. Price: \$7,995.

Brightware Inc., 101 Rowland Wy., Ste. 310, Novato, Calif. 94945, (800) 532-2890, (415) 899-9070, fax (415) 899-9080.

CONSYST SQL DESIGN INC. Circle No. 105

SQL Design 2.1 is a rapid client/server application development generator for SQL databases, and it offers database-independent cross-platform solutions. This 4GL tool uses a graphical user interface to create powerful, user-friendly programs cost-effectively. An efficient productivity tool for the development and maintenance of applications, SQL Design enables programmers to generate systems easily and in less time. Price: \$250.

Consyst SQL Inc., 15 Mont-Royal W., Ste. 120, Montreal, Que, Canada. H2T 2R9, (514) 849-7431, fax (514) 849-8125.

DSOFT DEVELOPMENT INC. Circle No. 106

dbfLIB Programmer's Library 2.04 is a C/C++ programmer's library for accessing dBASE data files and indexes. Included are royalty-free DLLs for Windows, Windows NT, and OS/2. Additional support for DOS is also included. dbfLIB provides a consistent set of APIs across DOS, Windows, NT, and OS/2 platforms.

This allows development of applications that can share dBASE programs currently in use, in addition to other programs that use dBASE files. This can all be done without migrating megabytes of dBASE files to another database environment. Price: \$195.

dbfREXX for OS/2 2.04 provides dBASE file access from your OS/2 REXX programs. Works with interpreted REXX or many of the popular REXX compilers. dbfREXX provides simple, affordable database management to your REXX programs. As a migration tool, many corporate developers use dbfREXX to move data from the mainframe to the PC environment for use as dBASE files. Price: \$99.

dSOFT Development Inc., 4710 Innsbruck Dr., Houston, Tex. 77066, phone/fax: (713) 537-0318

HILBERT COMPUTING Circle No. 107

Chron 4.0 is a 32-bit multithreaded PM application that will schedule OS/2, DOS, or Windows programs running under OS/2 at a selected time at selected intervals. Chron uses REXX as a macro language to extended the base scheduling options. Price: \$99.

Hilbert Computing, 1022 N. Cooper Ct., Olathe, Kans. 66061, (913) 780-5051, fax (913) 829-2450.

INTELLIGENT OBJECTS CORP. Circle No. 108

SQL Objects 2.2 will supply cross-platform, multi-database development tools. Encapsulated are all of the popular databases into a C++ class library. SQL Objects supplies drivers for Oracle, Sybase, SQL Server, SQL Sybase, Watcom SQL, Informix, DB2/6000, DB2/2, DDCS/2, Netware SQL, Btrieve, and so on. Cross-platform development is possible. Price: \$699.

Intelligent Objects Corp., 47 Stonewall St., Cartersville, Calif. 30120, (404) 382-6585, fax (404) 382-6374.

MACHINE INDEPENDENT SOFTWARE CORP. Circle No. 109

CQL++ 8.0 provides a single user or client/server SQL database engine including complete C++ source code. CQL++ is usable in any environment where a C++ compiler is available meeting the proposed ANSI C++ specification (templates not required). CQL++ implements ANSI SQL 1989 with ODBC extensions and provides

an embedded interface as well as an ODBC driver. Price: \$495 for a single-user, \$995 for client/server.

Machine Independent Software Corp., 1819 Ridgmont, Columbia, Mo. 65203, (800) 823-8672, (314) 446-4242, fax (314) 446-4243.

MERCURY INTERACTIVE Circle No. 110

WinRunner for OS/2 3.5 provides productive solutions for automated testing of GUI client applications. It tests the application automatically with speed, accuracy, and repeatability, thereby greatly reducing testing time. Price: \$4,995.

Mercury Interactive, 470 Potrero Ave., Sunnyvale, Calif. 94086, (408) 523-9273, (408) 523-9911.

OBJECT DESIGN INC. Circle No. 111

ObjectStore 4.0 is a high-performance database for storing complex data structures and nonrecord-oriented information in heterogeneous client/server environments. ObjectStore features distribution and transaction management for concurrent access to objects among members of a workgroup. It also offers state-of-the-art database functions such as data integrity, version control, query processing, and a complete multi-client/multi-server architecture. Price: contact vendor.

Object Design Inc., 25 Mall Rd., Burlington, Mass. 01803, (617) 674-5162, fax (617) 674-5262.

ON-LINE DATA Circle No. 112

OnCmd xBase for OS/2 1.1.5 harnesses the power and the speed of OS/2, enabling you to develop new or migrate existing applications from dBASE, FoxPro, or Clipper to native 32-bit GUI application environments. It comes with screen painter, report writer, DLL/DDE support, 350+ functions/commands, and is client/server ready. Unlimited run time and server add-on licensing is available. Price: \$249.

On-Line Data, 5 Hill St., Kitchener, Ont, Canada. N263X4, (519) 579-3930, fax (519) 579-2130.

PARCPLACE SYSTEMS INC. Circle No. 113

VisualWorks 2.0 is a powerful client and server tool for building portable applications using object-oriented technol-

ogy. A Database Application Creator is included for rapid application development. Applications developed are instantly portable across the enterprise and can have their functionality distributed between both client and servers. Price: \$2,995.

VisualWorks Report Writer, an extension to the VisualWorks product line, is a comprehensive query and report-generation tool for VisualWorks developers and users. Fully integrated with VisualWorks, this client/server development tool builds portable applications with object-oriented technology and provides point-and-click access to a number of reporting capabilities. With its spreadsheet-like macro language, ReportWriter is intuitive enough for users, yet powerful enough for programmers. Price: \$375.

ParcPlace Systems, Inc., 999 East Arques Ave., Sunnyvale, Calif. 94086-4593, (800) 759-7272, (408) 481-9090, fax (408) 481-9095.

POET SOFTWARE CORP. Circle No. 114

POET 3.0 is a C++ Object DBMS that supports encapsulation, inheritance, and polymorphism. It is available for 14 different operating systems and comes as single-user or client/server version. POET 3.0 features include OLE and ODBC support, schema versioning, check-in/check-out, online backup, as well as a developer workbench that includes class and object browsers. Price: contact vendor.

POET Software Corp., 999 Baker Wy., Ste. 100, San Mateo, Calif. 94404, (415) 286-4640, fax (415) 286-4630.

PROGRAMART CORP. Circle No. 115

APMpower Application Performance Analysis System is used to analyze MVS application performance data. Operating with Programart's STROBE Application Performance Measurement System, APMpower enables developers to interactively manage measurement sessions of MVS applications and to analyze their results using OS/2 workstations. Analytical aids, including application performance hints, online help, a database of module descriptions, and guided analysis methods empower developers, as well as technical staff to resolve application performance issues. The SQL analysis feature helps pinpoint SQL inefficiencies by suggesting specific database definition and coding changes that

could improve the performance of DB2 applications. Price: contact vendor.

Programart Corp., 124 Mt. Auburn St., Cambridge, Mass. 02138, (617) 661-3020, fax (617) 498-4010.

SAS INSTITUTE INC. Circle No. 116

SAS System 6.0 is an integrated suite of information delivery software for business decision making. It provides organizations with tools to access, manage, analyze, and present data within an applications development environment. Capabilities include EIS, data warehousing, client/server computing, database access, applications development, graphics, data analysis, report writing, quality improvement, project management, computer performance evaluation, and decision support. Price: \$985.

SAS Institute Inc., SAS Campus Dr., Cary, N.C. 27513, (919) 677-8000, fax (919) 677-8123.

SOFTWARE AG Circle No. 117

Natural 2.1.3.4 is a cross-platform fourth-generation language with the ability to support text-based or GUI development. Price: \$900-\$1,495.

Software AG, 11190 Sunrise Valley Dr., Reston, Va. 22190, (703) 391-6981, fax (703) 391-8290.

XDB SYSTEMS INC. Circle No. 118

XDB Workbench 4.0 enables mainframe DB2 development and maintenance to be off loaded to less-costly, more productive platforms such as PCs and LANs. Providing programmers with a DB2 equivalent environment to develop, test, and execute complete DB2 application on a desktop, XDB Workbench consists of a fully DB2 compatible database engine, a COBOL or PL/1 precompiler, and a suite of development and administrative tools. Price: \$1,500.

XDB Server 4.0 is a fully DB2 compatible database system that brings the power mainframe database systems to client/server platforms. XDB Server can be used as a repository for host-targeted development and as a production platform for running DSS or OLTP applications. XDB Server is currently available for OS/2. Price: contact vendor.

XDB Systems Inc., 9861 Broken Land Parkway, Columbia, Md. 21046, (410) 312-9300, fax (410) 312-9505.

An Internet-based Directory to Thousands of Products and Services



OS/2® & AIX

developers,

systems

administrators,

and users: the

tools you need

are just a few

keystrokes

away.

Thousands

of product &

service listings for

IBM supported

operating systems,

continually

updated, easy

to search.

<http://www.mfi.com/softwareguide>

Sponsored by IBM®, OS/2 Developer, OS/2 Magazine, and Software Development.

OS/2® is a registered trademark of International Business Machines Corporation and is used by Miller Freeman, Inc. under license.

Find It on the Web!



REXX + dBase = RexxBase

The dBase Database Access For REXX

- Works with DBF, DBT, NDX and MDX files.
- Supports dBase III and IV file formats.
- Uses standard REXX API function calls.
- Comes with a GUI Front-End for Database Maintenance.

Find the shareware version RXBAS203.ZIP on your favorite BBS.

Use it with VisPro/Rexx, VX Rexx and GpfRexx

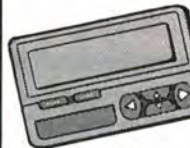
American Coders, Ltd. Post Office Box 97462 Raleigh, NC USA 27624
(919) 846-2014

Internet: joe@usacoder.rtp.nc.us CIS: 74150,2370

Only \$95.00 + shipping & handling

Circle Reader Service Number 22

Send text messages to pagers directly from OS/2!



- Easy-to-use workplace shell application
- Includes a command line interface
- You can 'page-enable' your applications!
- 32-bit, SOM-based code - Only \$79

ChipChat® Wireless Communicator

ChipChat-Cawthon Software

Dearborn, Michigan USA & Fukuoka Japan

Phone 313-565-4000 Fax 313-565-4001

Circle Reader Service Number 23

TCP/2™

TCP/IP for OS/2®

Our customers say our tcp/ip is the best because it's faster, more complete, and reliable.

Find out for yourself!

Supported: all released versions of OS/2 including 32bit API for 2.x and above; SLIP and/or PPP on COM1 and/or COM2; Ethernet; Token Ring; ARCNet; NDIS and ODI

(800) OS2-TCP2

Essex Systems, Inc.

One Central Street, Middleton, MA 01949

A Dan Lanciani Product

TCP/2 is a Trademark of DLD consulting. All other products and trademarks are the property of their owners. TCP/2 is Based in part on work done by the University of California Berkeley.

Circle Reader Service Number 24

FAX Developer Tools

From the developers of *FaxWorks for OS/2*

Client/Server API and Printer Driver Toolkits

Support for LANs, up to 96 lines per CPU, and all popular fax hardware

Keller Group Inc.

Voice: (612) 429-7273

Fax: (800) 329-3293

Email: kgroup@ibm.net

Fax: (612) 653-1987

Faxworks is a trademark of Global Village Communication.

Circle Reader Service Number 25

Opt-Tech Sort/Merge

High Performance Sort/Merge/Select Utility

Use as a stand-alone utility or call as a Subroutine. Many features including unlimited file size, multiple keys, record selection, duplicate elimination, summing and more!

Call for free brochure

Opt-TechData Processing
P.O. Box 678

Available for:

OS/2 or Unix \$249

DOS or Windows \$149

Zephyr Cove, NV 89448

Phone (702) 588-3737

Fax (702) 588-7576

Circle Reader Service Number 26

The easiest to use, most powerful 32 bit OS/2 PM IPF Language Editor...

New! IPF Editor 2.0

SRP:
\$195

- Designed for intuitive use by anyone
- Supports all IPF Tags, with simple index creation, easy multiple-window creation, and drag-and-drop panel ordering
- Generate C program and Resource source to add help to applications in minutes
- Import WordPerfect, Describe and RTF
- Preview IPF output without compiling
- Create all hypertext links automatically
- Sound Support and Spell Check
- Complete on-line help and documentation
- Optional multi-user network version

Orders:

1-800-IPF-7622

Information: (360)428-5025
on compuserve at 70410,2416

Perez Computing Services
4725 Monte Vista Place
Mt Vernon, WA 98273

Circle Reader Service Number 27

JOB SCHEDULING SERVER

OS/2 PM 32-bit advanced job scheduler for NOVELL and IBM LANs. Schedules multiple OS/2, DOS, and Windows jobs for concurrent execution on specified or auto selected LAN nodes. Conditional job scheduling (depends on files and other jobs return codes), security, sophisticated calendar, job logging, priority, user APIs, job recovery, periodic scheduling, and much more.

Microwork, Inc.

Phone: (708) 940-8979

Fax: (708) 940-8979



Network Remote Control And Monitoring

Help Desks Automated Agent Monitoring

PM/Pirate

Named Pipes
TCP/IP
Lowest network load
Record and Playback
All PM Video Modes
No Special Drivers
FAST!
Automated Cycling
\$89.00 + s&h



Digital Communications Solutions
2050 North Towne Ct. NE Suite 7
Cedar Rapids, IA 52402

Softtel also provides:
In and Outbound call center applications -
Voice information systems
Predictive dialing
Voice Mail
Internet Service provider and Client Software.
Presentation Manager Development and Consulting

(800) 701-7205

Circle Reader Service Number 29

TRACK 'em DOWN!™



Track:
• Bugs, Change Requests
• Projects, Code Changes
• Customer Calls and more

Customize:
• Reports
• Queries
• Forms & fields

Automatic Notification: Nothing falls through the cracks
Integration with Version Control: Link files to bugs & bugs to files
Multiple Linked Databases: Track bugs, customers and more
Industry Standard Databases: Access from other applications

SOFFRONT
Software Inc.

FREE working DEMO!

1-800-763-3766 / 408.263.2703

238 S. Hillview Drive, Milpitas, CA 95035 Fax: 408.263.7452

Circle Reader Service Number 30

Jagre SmartLink and Lotus Notes: A Winning Combination!

File Open

File Save

E-mail

Fax

Pager

OCR

Audio

Video

iRetriever

Help

In today's computer world, communicating effectively and quickly could be the difference between your company being a dead company or being a quick company.

Using Lotus Notes within your company, gives your company the ammunition it needs to be a quick company.

Now, Jagre, Inc. provides your company with the competitive advantage to being -- and staying -- a quick company: **Jagre SmartLink**.

Jagre SmartLink is a set of Lotus Notes-enabled tools grouped together in one place to empower your IBM OS/2-based desktop applications to communicate effectively and quickly with Lotus Notes for Document Management, E-mail, Fax, Pager, OCR, Voice, Video, and User-based Agents to gets tasks done quickly.

With Jagre SmartLink, the powerful features of Lotus Notes are now a button click away!

Click on any Jagre SmartLink tool for instant access to Lotus Notes. Now, you can use Lotus Notes and the Lotus Notes Companion Products with your desktop applications to communicate quickly with someone down the hall or across the world!

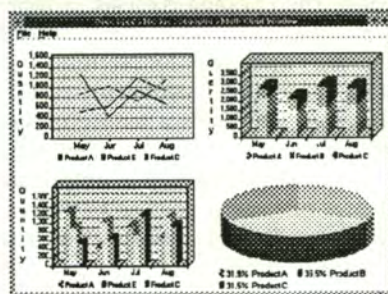
Now, your ammunition arsenal is complete with Jagre SmartLink and their arsenals are not!!

Jagre SmartLink

For a free demonstration copy of Jagre SmartLink to discover how it can help your company be a quick company, not a dead company, call or write Jagre, Inc. at (617)424-8302, 102 Gainsborough St. #202E Boston, MA 02115.

Circle Reader Service Number 31

Jagre SmartLink



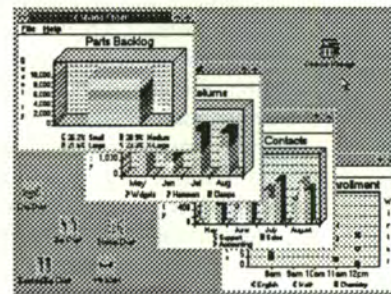
It's easy
to integrate
business graphics
into your
OS/2 PM[®]
or
Windows 3.x[®]
application

Developer's Business Graphics Toolkit

- High-level APIs
- Tactile feedback
- Real-time charting
- 16-bit and 32-bit
- Dynamic Link Library
- Positive/negative data
- Printer/metafile support
- 25+ chart types



The Crossley Group
P.O. Box 921759
Norcross, GA 30092
USA (404) 751-3703
Int'l 44-932-844-281
(demo available)



Circle Reader Service Number 32

Introducing...

the OS/2 DEVELOPER NOTES
Products and Services Guide

Each issue of OS/2 Developer will now feature this special advertising section.

Costs for an 1/8 page ad range from \$350-\$450

Costs for a 1/4 page ad range from \$600-\$900

Call For Details:

Kristin Morgan
212-626-2498

Advance at the speed of light.

Object-Oriented
Client/Server

DESIGNERS • DEVELOPERS • ARCHITECTS

Since 1979 Spectrum Concepts has enabled its clients to effectively meet their business objectives by providing leading-edge information technology consulting expertise. The continued success of our rapidly expanding and highly skilled consulting team has created new opportunities for talented professionals with the following skills:

- OS/2 GUI
- C++, OOD/OOP

Contact: Karin Lundgren (212) 791-4800 x250 Fax: (212) 791-6639
CompuServe: 74674.3571 Internet: spectrum@pipeline.com and/or mail resume to:
Spectrum Concepts 150 Broadway, Suite 600, Dept. OS2, NY, NY 10038



Circle Reader Service Number 33



New Products for OS/2

C-Vision 4.0. This set of tools allows programmers to analyze, understand, document, and maintain C and C++ source code. The package consists of four main components: a cross-referencer, a diagrammer for function-call graphs and class hierarchies, a source code reformatter, and an intelligent source code lister. The cross-referencer produces a listing of symbols enhanced with usage and type information, while the tree diagrammer provides numerous output formats for different views of your code. The C-Vision reformatter provides for flexible indentation control, including independent positioning of curly braces, nested preprocessor statements, case/default statements, and labels.

Gimpel Software Circle No. 180
Phone: (610) 584-4261, Fax: (610) 584-4266

CAPWatch 1.0. Creative Assistance Software announces the release of CapWatch 1.0, a new addition to its line of OS/2 LAN Server utilities. CapWatch is designed to meet the need of LAN Server administrators, particularly those using LAN Server Entry to track server utilization against configured capacity parameters. In addition, this product provides real-time graphing of the utilization, logging of the data, graph capture, and re-display and configurable alarm and alert levels. The program may be operated locally on the server being monitored without generating network traffic, or remotely across the LAN from any OS/2 workstation.

Creative Assistance Software Circle No. 181
Phone: (704) 544-0001, Fax: (704) 543-6366

BridgeWare. The BridgeWare family of products is the first suite of development and run time tools to provide access from GUI fourth-generation languages, such as Powerbuilder and VisualBasic, to Micro Focus COBOL, and CICS applications. This product generates a complete interface from 4GL applications to COBOL with no manual coding, rewriting, or retesting required. It extends 4GL client/server applications with true multitasking, server-based business logic for development and deployment on a wide range of environments. The BridgeWare family of products includes Micro Focus DeskTop BridgeWare and the

Micro Focus BridgeWare Server Enabling Kit, both suited for rehosting and partitioning COBOL and CICS applications for reuse within a client/server architecture.

Micro Focus Circle No. 182
Phone: (415) 856-4161, Fax: (415) 856-6134

Gui-Kit 1.1. This cross-platform GUI toolkit for C and C++ simplifies the development of GUI-based applications. Gui-Kit provides an intuitive object-oriented programming environment for C or C++.

Visual Systems Corp. Circle No. 183
Phone: (612) 434-6382, Fax: (612) 434-6538

LinkWiz. This product makes file transfers fast with graphical interfaces that operate under OS/2 2.1, OS/2 Warp, and OS/2 Warp Connect. Operating across platforms and in multiple environments, LinkWiz transfers files from one computer to another. The utility software also features file and disk management capabilities and provides high-speed data-transfer with build-in error correction, moving up to 8 MB a minute over special parallel and serial cables that come with the software.

PCX Circle No. 184
Phone: (619) 481-6300, Fax: (619) 481-6474

OOMetric 2.0. This object-oriented metrics tool now calculates and reports on over two dozen object-oriented metrics for class, size, class inheritance, class internals, project concerns, method size, method internals, and method externals. Both project and design metrics are evaluated. Design metrics cover all aspects of both the quality of the object model and the most efficient implementation of its architecture. Project metrics handles administrative concerns, estimating and reporting on scheduling and staffing during the development process. OOMetric is a standalone program with probes for a variety of popular Smalltalk dialects.

Hatteras Software Circle No. 185
Phone: (919) 319-3816, Fax: (919) 319-3877

A few words from our competitors:

Prtablty.

Prtbilty.

Portblty.

Portabit.

No matter how they misspell it, it's still not portability. At Zinc, we understand that porting the last 20% of your code takes 80% of your time.

Only Zinc offers complete portability.

With Zinc® Application Framework™ you can develop on the platform you prefer. And since Zinc is the only one that delivers 100% portability, you'll have your application on other platforms as fast as you can recompile. It's part of what makes Zinc the most productive—and affordable—tool a programmer can use.

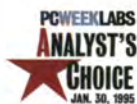
Productivity that leads to opportunity.

Portability is just one road you'll find a little easier. Zinc zips through tedious tasks with C++ object orientation and a unique visual development tool. Globalizing your application is as easy as translating the text. And Zinc is the only product that supplies 100% of the source code.

It all adds up to productivity. Which means more profitability. Which is a concept we probably don't have to spell out for you. For free information and demonstration software, call toll-free:

800 638 8665

Outside the U.S. call: +1 801 785 8900. In Europe call: +44 (0)181 855 9918. In Asia: +81 (052) 733 4301. Contact Zinc electronically at info@zinc.com or GO ZINC on CompuServe.



INFOWORLD
February 6, 1995

"Developers seeking easy delivery of GUI applications on DOS, Windows, OS/2, Macintosh and Unix platforms, or pursuing international markets will find Zinc their best option by far."

"Zinc came closest of all the products we tested to our ideal of portability: just copy the code to the target machine, then recompile and relink the application...In short, Zinc did a great job."

z i n c

NO LIMITS

The verdict is: One's a pleasure, two are useful, and one's a pain.



Scores run from 7.2 to 2.7, from worthy to worthless.

	zApp Developer's Suite	C++View 3.01	zApp Application Framework 4.1	zApp
✓ Installation & Configuration (7.5)	46.87	46.87	37.50	37.50
✓ Generating a Skeleton App. (12.5)	125.00	31.25	62.50	31.25
✓ Interface Design (15.0)	112.50	75.00	93.75	0.00
✓ Event Processing (12.5)	78.12	78.12	62.50	62.50
✓ Enhancement & Maintenance (12.5)	93.75	78.12	78.12	31.25
✓ Portability (15.0)	75.00	75.00	112.50	37.50
✓ Documentation (10.0)	62.50	25.00	62.50	0.00
✓ Support (10.0)	100.00	100.00	75.00	50.00
✓ Pricing (5.0)	31.25	31.25	50.00	25.00
FINAL SCORE	7.2	5.4	6.3	2.7

Dr. Dobb's JOURNAL

There's a saying that good design is invisible. As Ward Cunningham puts it, "Good class libraries whisper the design in your ear." This is the case with zApp, which straightforwardly provides all the usual classes you'd expect in building event-driven GUI programs: a main application class, a small hierarchy of event handling classes, classes for graphic display, window-containing classes, and the usual GUI widgets (push buttons, check box, list box, and so on). There are no radical concepts or unpleasant surprises, greatly easing any learning curve associated with a brand-new API.

Ray Valdés - October, 1992

Program NOW

The documentation provided with zApp is little short of superb. zApp 2.0 is a professional product which turns platform-independence into a practical proposition by its efficient, versatile and function-rich implementation.



When presenting zApp with their Editor's Choice Award, Willie Watts explained why he "plumped for zApp": Of all the C++ encapsulations of Windows in all the bars in the world, we both felt this was the best attempt we had seen, and was something of a landmark in the use of this important language.

JOURNAL OF OBJECT-ORIENTED programming

The class design is the most important and most subjective aspect of a class library, and I think zApp's is superb. zApp encapsulates the entire underlying native API. In other words, if you want a Windows dialog, you get a Windows dialogue, not something that looks like one.



The 1992 Star Tech Award for C/C++, and Application Frameworks goes to Inmark Development Corp's zApp V.2.0. zApp is simply the best designed application framework on the market. Its well-conceived classes make Windows programming an easier and more manageable process-and even fun. If C++ is your language of choice, then you should be using zApp. Period.



For general-purpose applications that aren't limited to text, basic graphics, and dialog boxes, however, I suggest evaluating zApp first. Features like advanced graphics, printing support, DDE, compatibility with third-party resource tools, and the ability to incorporate custom controls can be hard to live without...

Reprinted from "COMPUTER SHOPPER" Sept. 1993
Copyright© 1993 Ziff-Davis Publishing Company.



As long as you write your applications using zApp objects, you can create programs for one compiler or environment and recompile them with another compiler for a different environment without any changes... zApp's object hierarchy remains the most comprehensive I've yet seen in an application framework.

Reprinted from "PC MAGAZINE" Dec. 22, 1992
Copyright© 1992 Ziff-Davis Publishing Company.

C++REPORT

Marketing people say things like "zApp is great. It contains over 200 classes." That looks good in a slick and glossy brochure, but it leaves the fundamental question unanswered: Are they good classes? They are. zApp's classes are well-implemented and easy to use. The library covers the full breadth of contemporary functionality and offers portability across most major operating systems.



What I've noticed about the company behind the product is that Inmark's customer technical support is outstanding (usually a returned call within four hours), their BBS is full of great information, and their attention to quality is stellar. I wish all software tool providers did as well.



Mark Brittingham, a user-interface specialist with AT&T Bell Labs in Middletown, NJ, chose zApp over other frameworks. "You can see it's built by someone who builds applications,"

Reprinted from "PC WEEK" July 19, 1993
Copyright© 1993 Ziff-Davis Publishing Company.



With the addition of a visual screen builder, the final barrier to zApp productivity has fallen. For quick, form-based applications, zApp Factory makes C++ productive enough to be competitive with the interpreted languages. For more serious applications, the zApp framework is a superior application framework. I would use it even if I were just programming for Windows, but the fact that it compiles for Windows, Win/NT, OS/2, UNIX, and X/Motif, is a nice little bonus.



zApp

Award winning developers use Award winning tools.

With the award winning zApp® Developer's Suite, you can quickly build state-of-the-art C++ applications that run across fourteen of today's most popular operating systems. The zApp Developer's Suite contains the latest in drag-and-drop design, prototyping and testing, code generation, and powerful GUI objects. Best of all this powerful technology sits on top of zApp, the industry leading application framework in use by tens of thousands of developers worldwide.

For a free demo call: 1-800-346-6275



Inmark Development Corporation 2065 Landings Drive, Mountain View, CA 94043 (800) 346-6275, (415) 691-9000 Fax: (415) 691-9099

In the U.K. and Scandinavia, call PTS/Software Plus at +44 (0) 928 579900. In France, call PTS/Software Plus at (05) 908194. In Germany, call ESM Software at 07702-9256-0. In Italy, call Silicon Valley On-Line at (049) 8719820. In Australia, call Micro Way at (03) 580-1333. BBS: 415-691-9990 • Internet: info@inmark.com • CompuServe: GO INMARK